

ACCESSION FOR	
CFSTI	WRITE SECTION <input checked="" type="checkbox"/>
DRG	BUFF SECTION <input type="checkbox"/>
UNCLASSIFIED	<input type="checkbox"/>
JUSTIFICATION	
File 423	
BY <i>fm</i>	
DISTRIBUTION AVAILABILITY CODES	
DIST.	AVAIL. END OF SPECIAL
/	

SP a professional paper

Breaking the Cost Barrier
in Automatic Classification

by

L. B. Doyle

1 July 1966

SYSTEM

DEVELOPMENT

CORPORATION

2500 COLORADO AVE.

SANTA MONICA

CALIFORNIA

90406

The research reported in this paper was partially supported by Contract AF 19(628)-5166, Prototype Library Project, for Rome Air Development Center.



TABLE OF CONTENTS

	<u>Page</u>
I. ATTITUDES TOWARD AUTOMATIC CLASSIFICATION	3
A. <u>Technical Problems of Automatic Classification</u>	4
(Brief description of the cost barrier and a statement of objectives)	
B. <u>A Special Problem: Intellectual Resistance</u>	4
Five "cases against automatic classification"	5
1. Word-similarity classification not optimally useful	6
2. Classification not necessary when retrieval feasible	7
3. Users should have tailormade classification	8
4. Subject access is not much used in many libraries	8
5. Classification is more than grouping by topic	9
A comment on the influence of computational linguists	10
II. THE NATURE AND POTENTIAL OF SOME NOW-FEASIBLE CLASSIFICATION METHODS.	11
A. <u>Topical Relatedness and Word-Content Similarity</u>	13
How topical closeness relates to percentages of words in common	15
Why similarity grouping is more accurate among topically close documents than among the topically remote	16
B. <u>How a Classification Procedure Based on Word Similarity Operates</u>	16
(Example: the Ward grouping program)	
C. <u>Advantages of a Cluster Analysis Method Over the Ward Grouping Program</u>	18
Consequences of a relationship between classification accuracy and "amount of information" used in classifying	19
D. <u>Topical Profiles</u> (Example: word profiles in a public library)	20
E. <u>Iterative Classification Using Profiles</u>	23
Use of weighted scoring to tap the total information content of a profile	24
The process of convergence to maximum similarity clusters	26
Computer time usage in classifying 30 million items	27
F. <u>The Potential of Large-Scale Qualitative Cluster Analysis</u>	29
Problems of individuals and governments in coping with bigness	30
Three imagination-stretching examples of possible application of qualitative cluster analysis on a very large scale	31
1. What-to-invest-where analysis in agricultural planning	31
2. Extrapolation-of-experience analysis in health and medicine	32
3. Sorting-the-attributes analysis in juvenile delinquency	33

	<u>Page</u>
III. THEORIES, FEASIBILITY TESTS, AND PROJECTIONS	35
A. <u>The "Why" of Lists and Profiles</u>	36
Why frequencies are used with profiles but not with lists	37
The compromise between total-information usage and clustering tendency representation in deciding profile weighting	38
B. <u>The Quest for Convergence</u>	40
Specific problems	
1. Premature convergence: metastability	41
2. Cyclic instability: three kinds of "sloshing"	41
3. Indefinite cluster boundaries: ambivalence	43
C. <u>The Attainment of Convergence and a Novel Demonstration</u>	43
Finding the inner clusters: the key to rapid convergence	44
A demonstration of cluster reproduction involving the reader	45
D. <u>Prospects for Declining Costs and Continued Development</u>	58
REFERENCES	62

1 July 1966

1
(page 2 blank)

SP-2516

ABSTRACT

A low-cost automatic classification method is reported that uses computer time in proportion to $N \log N$, where N is the number of information items and the base is a parameter. Some barriers besides cost are treated briefly in the opening section, including types of intellectual resistance to the idea of doing classification by content-word similarity.

The second section explains the basic processes of document grouping by similarity, and discusses the advantages of the reported method over methods commonly experimented with. The operation of an iterative procedure using word profiles to progressively improve the grouping of content-word lists is described. Then some possible applications aside from document classification are enumerated.

The final section begins by presenting theoretical underpinnings that explain the form taken by the components of the method. An account of the struggle to make the method work is sketched, followed by a cycle-by-cycle description of a feasibility demonstration. The conclusion states that mere cheapness is not enough and analyzes what researchers and developers might have to do before user acceptance of automatic classification can be assured.

BREAKING THE COST BARRIER IN AUTOMATIC CLASSIFICATION

"We begin with what seems like a paradox. The world of experience of any normal man is composed of a tremendous array of discriminably different objects, events, people, impressions. There are estimated to be more than 7 million discriminable colors, and in the course of a week or two we come in contact with a fair proportion of them. No two people we see have an identical appearance and even...the same object over a period of time changes appearance...with alterations of light or in the position of the viewer...for human beings have an exquisite capacity for making distinctions.

"But were we to utilize fully our capacity for registering the differences in things and to respond to each event encountered as unique, we would soon be overwhelmed by the complexity of our environment. Consider only the linguistic task of acquiring a vocabulary fully adequate to cope with the world of color difference! The resolution of this seeming paradox--the existence of discrimination capacities which, if fully used, would make us slaves to the particular--is achieved by man's capacity to categorize..."

—A Study of Thinking (1)

I. ATTITUDES TOWARD AUTOMATIC CLASSIFICATION

"Automatic classification" is a way of applying digital computers that might be more descriptively termed "programmed organization of complex nonquantitative data." Growing out of research in use of computers to aid document retrieval, methods of grouping and organizing text items according to content-word similarity have not only reached a point where they might revolutionize computer capabilities in natural language processing, but are easily extendable to any large collection of data consisting of identifiers rather than measurements.

As examples, a list of index tags or key words of a document could as easily be a set of event descriptors (accidents, crimes, overseas happenings), a five-year record of symptoms (preventive medicine), or a list of trace elements and components in a soil sample (agriculture). In all cases the capability to group in highly organized form an enormous number of qualitatively described individuals, objects, samples, etc.,

is a new kind of computer application whose potential is limited only by the imaginations of possible users.

A. Technical Problems of Automatic Classification

Automatic classification as applied to natural-language data has had its fair share of technical hurdles before it came to deserve being called a usable tool. Progress has been much less difficult than for machine translation, but on the other hand its problems have been substantially more difficult than those of automatic concordance-making. Actually, the problem of just doing automatic classification in ways comparable to and probably exceeding human performance has been solved for document collections of no more than a few hundred (2,3).

But one significant aspect of it has been resisting solution: cheap classification for large numbers of items. In document classification it is widely understood that existing methods of cluster analysis or grouping by similarity are not economical for collections exceeding 10 or 20 thousand documents; these methods are troublesome primarily because they require generation and processing of a "similarity matrix," which reflects the index tag or content word commonality for every possible pair of documents in a collection, and the size of the matrix of course increases as the square of the size of the collection since there are $N(N-1)/2$ possible pairings of N items. This and other factors cause computer time to be consumed in proportion to the square or even to the cube of the number of items to be classified. Only where a priori classification criteria are employed have people been able to avoid this square proportionality, but the use of such criteria is a significant departure from automaticity of classification.

This document reports on a method of automatic classification that uses computer time in direct proportion to the number N of items, or--strictly speaking--in proportion to $N \log N$, where the logarithmic base is probably greater than 20. Section I introduces the topic and describes ambient attitudinal sets that could weigh even more than technological considerations in rate-limiting the method's application. Section II sketches the nature of the method and what it is capable of doing, with both prosaic and exotic (though not unlikely) examples. Section III discusses theoretical foundations, presents a technical feasibility exercise, and makes some concluding remarks on cost.

B. A Special Problem: Intellectual Resistance

We note at the outset that automatic classification is not merely a technically difficult problem. This researcher, after having worked on a variety of problems from nuclear reactor safety to statistical aids for analyzing English grammar, has found automatic classification to

be unique in its tendency to provoke skepticism. Some of this skepticism reflects an honest interest in the problem, but so much of it is outright negativism that it is quite probably exerting an unnecessary drag on progress. Indeed, there is a danger that the prevalence of these states of mind can prevent or hinder many applications of automatic classification even after it has been shown to be technically and economically feasible for those large-scale data processing situations most needing its complexity-reducing power.

Therefore, there is no better place to take up this problem than in the pages to follow. Enumerated below are what are felt to be five of the most typical or influential of the arguments invoked against automatic classification. The positions held are often sophisticated and even correct, but are misused as arguments specifically against automatic classification; unfortunately, though the factual pertinence of the arguments is small or nil, the psychological impact is usually large, because most of those influenced have little reason not to be swayed by the admittedly impressive reasoning often involved. The five "cases against automatic classification" follow:

1. Classification of documents according to similarity of content words, even if impeccably done, does not equate to optimally useful classification; one reason for this is the practice of treating content words as equally significant, letting frequency counts decide which will be chosen, when actually the topical representativeness of words is unlikely to be related to frequency.
2. Classification is unnecessary in the computer age when one can do retrieval.
3. The computer makes it possible for each user to have document-reference organization tailor-made to suit his own information needs, but automatic classification as currently conceived imposes the same scheme on all users.
4. In big university libraries only a small portion--perhaps 5%--of the requests for information are according to subject.
5. A lot more is involved in classification than just grouping by similarity or even grouping by topic.

Each of these arguments will be met individually, since it is not to be denied that each has a fundamental meaning that has to be answered either by theory or by the pressure of events, even though each is misused as a specific argument. There is, however, an interesting way

of meeting them all simultaneously, by resorting to a loose but hopefully effective analogy. Imagine a researcher who is crusading for the study of water resources. Think of him as being opposed by the following arguments:

1. What good is water? I want soup in the winter and beer in the summer.
2. We Frenchmen drink only wine, haven't you heard?
3. Develop water resources? What are you, a Communist? In this country each of us digs his own well.
4. None of us turtles use liquid; we eat those luscious plants down by the Gila River.
5. What do you know about water? We rainmakers have been developing water resources for generations.

Notice that each of these uses a specific water system or a specific way in which water is utilized, as an argument against the development of general capability. This is precisely the kind of argumentation being directed at automatic classification, and the idea is seemingly rejected that general capability might one day be adapted to the uses of systems created. We may now be better able to appreciate how this applies in detail to the various positions held.

1. Word-similarity classification not optimally useful. This idea in itself is valid. Utility of grouping must surely vary according to information access requirements, and the groups of automatic classification are determined by word occurrence patterns, not by indexer judgment or user need. It is also quite sophisticated to recognize that an infrequent word, or even a word occurring nowhere in the document, may be more topically descriptive than a frequent word. What is not considered is that in the history of development of automatic processes some advantages often had to be sacrificed in the beginning to gain the benefits of automaticity; later on many of these were won back again.

As an example, the first automobiles sacrificed reliability, and it was the frequent breakdowns that led to the derisive slogan: "Get a horse!" Autos had difficulty at night and in bad weather, and roads were not adequate; close attention to control was required by the driver. But the advantages of high speed, power, and endurance proved so important that automotive engineering was soon compelled to overcome the disadvantages.

Just as the automobile was deaf, dumb, and blind, present computers must classify with no feeling for semantics or relevance. Such a

disadvantage is hardly a decisive one. Just as automobile builders made provision for driver control, developers of automatic classification systems can find ways for control by specialists or users. But one must know at the beginning that a hydrocarbon fueled motor will work and that power will be efficiently relayed to the wheels; if these things can't be achieved, there is no point at all in refining the brakes and steering. This consideration applies almost identically to automatic classification in its current state of development. To apply the automobile analogy still closer, we might picture 1964 as the year researchers in automatic classification were still trying to make their gasoline engine just revolve steadily; not until this was achieved would there be any point in trying to propel a vehicle with it. In 1965 they found they could drive around the block. By 1966, some have gone around the block in marathon fashion so many times that they can now look toward such things as going cross country and designing gas stations.

2. Classification not necessary when we can do retrieval. This recalcitrant major premise has profoundly affected the whole history of computer applications for document retrieval. Its influence is admittedly declining, but yet it is still strong enough to prevent many in the computer field--people who might be in the best position to do work in automatic classification--from straying from a "Don't look it up; ask the computer!" fantasy. The ease of constructing an electronic "penny arcade steam-shovel" that will pass as a retrieval system still tempts many who, though some of them are actually on the periphery of the document retrieval area, compete for funds with those trying to follow a more responsible and long-term approach.

Even though the "main stream" retrieval thinkers now recognize the need for man-machine partnership in searching, there still exists a curious denial by many that grouping facilitates searching. There is much talk about "browsing," and though this is a tremendous improvement over the "request terms with Boolean connectives" bind of 1955-1960, there is a lack of both ideas and projects relating to development of browsable formats. "Permuted title indexing" was made public in 1958, and there have been almost no basic improvements over that to beckon the eye of the browser.

Hardware technology, advancing at an impressive pace, almost seems to do more harm than good in reference to automatic classification, because the "powerful" new gadgets--display scopes with photoelectric pointers, on-line teletypes, rapid and flexibly accessed auxiliary storages, and so on--keep giving a new lease on life to the 1955-1960 concept of retrieval, typically: "Tell me what you're looking for, I'll keypunch some tags--with and's, or's, & not's--and the computer will search the tape and find it." Some of the people who are today building retrieval

systems around display scopes not only have not profited by the sad experiences of their predecessors, but in some cases don't even realize that their predecessors existed.

Display scopes, fortunately, do offer some hope of reversing the tide in favor of classification, because the almost instant response to a query by display of a two-dimensional and nearly page-sized format may sooner or later cause embarrassment as a result of the contrast between the sheer power of man-machine transfer of information and the dreariness of what is being transferred.

3. Users should have tailormade classification. This is an extension of argument #1, but is dealt with separately because it often involves a synthesis with argument #2, leading to a retrieval mode in which a user generates whatever organization suits his fancy at the time of search. With the qualification that the user may have difficulty knowing what groupings might benefit him or how to specify them, this synthesis may not be a bad idea. Furthermore, one can readily admit that yes, if possible, users should have tailormade classification.

But this reasonable idea is usually made to carry with it the implication that there is no use for other kinds of classification. As a general principle, this contradicts much of our experience, since it appears to say: possession of a custom-built facility renders public facilities of no value. Actually, civilization abounds with instances of systems structurable for individual needs coexisting with systems of standardized structure fulfilling the same general function. Standardized systems are usually cheaper, and certain aspects inherent in standardization also make them more convenient in many ways--think for a moment of the ease of having repair work done on Fords or Plymouths, in comparison to what was the case for foreign-made cars when they first became popular in the U.S. As another example, a car-renting traveler in Switzerland can make discoveries about the convenience of railway travel over private transportation that often astonish him.

4. Subject access isn't much used in libraries. Though former arguments are typically advanced by people in and on the fringes of the computer field, this one and argument #5 are more often heard from librarians, documentalists, and others in the actual practice of information service. It is of course possible to quarrel with the factual truth of this argument, even though it may apply to some university libraries. It is crucial, however, to see what is behind this argument.

It appears to claim that there is no point in working on automatic classification because "our experience shows" people won't use it. However, we shouldn't forget that a major reason people often don't use seemingly useful facilities is that of inconvenience. If certain

university libraries don't experience much access by subject, it may mean a lot of things: that access is easier, more opportune, and more informal elsewhere on the campus (e.g., asking the professor, inspecting citations, etc. and simply asking the library for the book by title and author); that measures to protect the library's collection make "access less accessible" (e.g., patrons having to check in books and briefcases before entering the stacks); or that effective reference tools cannot be afforded. Therefore one is glib, to say the least, if he leads directly from a set of use statistics to the generalization that something is inherently not useful.

There are reasons, having to do with their concept of their profession, why librarians would not be particularly overjoyed with the idea of automatic classification. The feeling is that classification is more for the benefit of librarians than for patrons, being literally a major component of their information-management system. But automatic classification aims toward aiding the literature-searching process, and is not (yet) concerned with helping librarians keep track of their holdings. The classification and subject cataloging practices in large libraries are worked out in such detail that it would be difficult to introduce capabilities that are radically new without much disorganization.

Appreciating this as one might, one is mistaken to use "how things are done" as an argument against developing basically new ways of doing things. We might well imagine, as a parallel, a society of the elite among telegraph operators in the 19th century discussing, in tones of disdain mixed with anxiety, the possibility of the proliferation of a new gadget known as a telephone--an instrument of remote communication that "practically anybody would be allowed to operate."

5. More is involved in classification than grouping by topic. This viewpoint is most likely to be expressed by the neo-classificationists, most of whom accept Ranganathan's impetus toward a much higher degree of conceptual systemization in classification than has been previously practiced. Such a neo-classificationist is likely to be appalled by a classification procedure having no conceptual regulation whatever, as is currently the case with automatic classification.

But intelligent and well-trained people, and hence, presumably, amply paid people are needed to master the intricacies of Ranganathan's kind of subject analysis. Even the more mundane forms of conventional classification are not cheap, and book-cataloging costs are notoriously high in today's libraries. The world is piled high with documented information that no one can afford to classify, and for which the arguments of the neo-classificationists are academic. Therefore they would surely do society a disservice to oppose on some intellectual basis a classification technique that could be orders of magnitude less

expensive than the cheapest conventional classification. Such a technique could eventually even help them lower the cost of their own kind of classification, making it more widely applicable.

The foregoing five arguments are not by any means the whole story of the "sociological" problems that have beset the development of automatic classification. For example, the opposition of linguists to "frequency methods" has not been discussed; this affects progress in automatic indexing and abstracting as well as in classification. Since the focus of the linguists is less specific, perhaps less ought to be said about their arguments. Moreover, the linguists have been dealt with adequately elsewhere (4).

This much is added, however, in commenting purely on the content of their arguments: they feel it is necessary to think about language in reference to certain well-worked-out modelistic frameworks; this is so strongly obligational that it has large consequence for everyone working in language processing and allied fields. In the fraternity of "computational linguistics," statistics is seldom accepted as a form of "computation," even though "computation" once meant "counting." Frequency of words and word structures is seen as either an irrelevant or an uninteresting medium of language analysis. Even when frequencies are used only as a means to an end, as in deciding what content words should be representative of a document, with no special significance attached to the values of the frequencies, the viewpoint still applies.

Computational linguists will seldom actually attack the statistical approach to analysis of text; most will simply ignore it. There are some, however, who are interested enough in that area to read the material reasonably carefully, although they themselves will not work in the area nor encourage others to do so. One linguist commented, "These numerical methods do not help me to think about language." In other words, so much valuation is placed on having a "phrase structure" or "transformational" type of language model, that computational linguists must stick to it not only in their thinking, but even in their processing operations. (A very few exceptions to the latter exist; for example, the machine translation project at RAND did from time to time explore word counts.)

The somewhat passive position against things statistical might not seem particularly menacing, but for the fact that a new generation of computational linguists is upcoming rapidly in numbers and in influence. A major orientation of this group appears to be, "There is no god but Sentence Structure, and Chomsky is his prophet." This new breed may well displace the assorted mathematicians and programmers in the computer field as the main obstructive force against the diversified approach required to assure maximal progress.

In comparison to the oncoming linguists and the still-influential, undifferentiated computer people trying their hand in the retrieval area, the practicing librarians and documentalists--despite their often more rankled tone--may not really have contributed much to the stifling of such routes as automatic classification. So perhaps the adage is true that a barking dog doesn't bite. Unfortunately, some people well outside of direct participation in the world of libraries and information services will, for whatever private reasons, borrow arguments like #4 and #5 as "expert opinion" to discourage whatever needs discouraging in their eyes.

There are a good number of instances of people with library backgrounds having attained positions in government and professional organizations where "thinking big" was called for, and having then become protagonists of some of the more imaginative research and development efforts in the computer field. We can be grateful that these dozen-or-so have acquired the influence they have, and--recognizing the need for computers in dealing with any and all kinds of documented information--have been reasonably free from dogma.

- - - - -

There are employees of metropolitan dailies who are in charge of the files of back issues and ever-bulkier morgue, and who have never heard of Ranganathan; and there are people at state hospitals maintaining files of case histories who do not know what computational linguistics is. They do know, however, that it costs from 10¢ to a quarter to process each item and maintain it on file. They sometimes wonder whether it is worth keeping a file when it is so much trouble to look up something according to other than its file heading. But the one in charge of the morgue, at least, doesn't know how lucky he is: he has potential access to machine-readable text. What would be his reaction if he were told that the value of his file could be doubled by making the items more accessible, and that this could be done--using a computer--for 1½¢ per item? Would he offer intellectual resistance? Yes, very likely. He would summon up all of his extensive experience and firm convictions and, looking squarely at his informer, would say: "Mister, I just don't believe it."

II. THE NATURE AND POTENTIAL OF SOME NOW-FEASIBLE CLASSIFICATION METHODS

The root purpose of automatic classification is to bring the human mind in contact with all or part of an information store. A large store of data in unordered files can be interrogated, but it cannot be grasped. It can be entered by specifying index terms or file numbers, but it cannot be entered from the top down (by going from the general to the

specific) nor from the bottom up (specific to generic). Category-subcategory relations may exist among the index terms, but this may not be representative of the structure of the file itself; an unordered file is an unordered file, and to the extent that it is made to conform to an order among elements external to it, it is no longer an unordered file.

As an example, a classification of biological species can exist on some chart or scheme external to a file containing information about animals, but only if the information items are made to bear all search-codes corresponding to the species, genus, family, order, etc., up to the phylum, does the file itself become ordered symbolically, which can be transformed to physical order by appropriate machines. Once this external hierarchy is impressed on the file, its structure can be grasped by reference to the external scheme, and it can be entered and searched in terms of its structure at any level of detail.

Order according to biological classification was used as an example because it has one further property of interest: it is about as far as can be from automatic classification. What makes automatic classification automatic is that the order of the file is derived from the native content of the file elements rather than from an external source. One could always quarrel with this by imagining the whole of the biological universe classified by some future race of robots, "who," by usual definitions, would be considered automatic agents.

This sort of objection, however, serves no purpose but to undermine a highly useful distinction: we like the word "automatic" to reflect a capacity for computers to discover whatever internal order is inherent in the information items themselves. If we consider the process of classifying in isolation, then the difference between automatic and what we might call quasi-automatic is reduced to the simplest of terms: the automatic classification program would consult nothing but the information items for class-generating criteria, whereas the quasi-automatic process would require a table of class data separate from the items, plus criteria for recognizing that a given item belongs to a given class. Notice that since we are viewing the classification process in isolation, we have no way of knowing how the class data were derived; as a methodological matter, it makes no difference whether these classes were derived by robots or by people.

That the above distinction is not an arbitrary one can be seen by reference to a "thought experiment." Suppose one falls heir to a keypunched Library of Congress classification schedule, and uses it directly for the classification of some documents whose keypunched text he also has on hand. From his viewpoint, all is easy and automatic: he reads in the classification schedule deck and then the text, presses the start

button, and the off-line printer is soon spewing forth the document assignments, arranged by author, by document number, and by the order of the schedule itself. He is tempted to call this "automatic classification."

In this description, however, we have overlooked one little intermediate step: obtaining the rules for deciding how a document is to be categorized. The problem is not just that the rules must initially be derived manually, by correlating judgmental classifications with unique features (presence of certain words or headings to an unusual extent) of documents in a given category, but that no guarantee exists that these rules will apply to documents outside of the collection for which they were derived. This sort of classification procedure could well be useful and labor-saving, but would require so much monitoring that we would have to consider it semiautomatic. In what we have termed automatic classification, one may still have reason to provide for human intervention and monitoring, but this is an option that depends on one's philosophy, and not a necessity to safeguard against gross errors. This follows from the basic difference in what the "a priori schedule" method and what the automatic method are trying to accomplish; the former aims at imitating human judgment, whereas the latter attempts to organize documents according to their "family resemblance" as manifest in their content words.

A. Topical Relatedness and Word-Content Similarity

The idea of word similarity as a determinant of classification needs to be understood, for it is the basic element in the classification method herein described. Section I, it turns out, can serve an additional purpose, aside from squashing the various arguments against this sort of classification. Section I was difficult to write, and four drafts were written before the author was satisfied with it; the first three drafts, however, were not immediately thrown in the trash basket, because it was recognized that they would serve as first-class illustrations of word similarity. The four drafts are about as "topically close" to each other as a set of documents could be; in published documentation, one would find (on some judgmental basis) this degree of closeness only in updated issues, revised editions, or condensed-for-publication versions of some starting document.

The content words of each draft were counted manually (which is not as difficult as it sounds), and the top 36 words in frequency of occurrence were selected; where ties existed in the neighborhood of the 36th rank word, the words occurring closest to the beginning of the section were chosen; suffixes were normalized in a manner that is known to be feasible on computers. The words for each draft are shown below, with the words listed in rank order:

<u>Draft #1</u>	<u>Draft #2</u>	<u>Draft #3</u>	<u>Draft #4</u>
CLASSIFICATION(37)	CLASSIFICATION(43)	CLASSIFICATION(53)	CLASSIFICATION(54)
ARGUMENT (23)	AUTOMATIC (22)	AUTOMATIC (27)	AUTOMATIC (28)
AUTOMATIC (16)	ARGUMENT (16)	ARGUMENT (17)	ARGUMENT (20)
WORD	COMPUTER	COMPUTER	COMPUTER
DOCUMENT	DOCUMENT	DOCUMENT	DOCUMENT
PEOPLE	ITEM	RETRIEVAL	WORD
INFORMATION	WORD	WORD	PEOPLE
ACCESS	PROCESS	PEOPLE	RETRIEVAL
---ANALYSIS	FREQUENCY	SYSTEM	LIBRARY
RETRIEVAL	SEARCH	USER	SYSTEM
COMPUTER	PEOPLE	LIBRARY	GROUP
ITEM	USER	GROUP	INFORMATION
SUBJECT	LIBRARY	---WAY	---WORK
---FILE	SYSTEM	INFORMATION	===LINGUIST
FREQUENCY	RETRIEVAL	METHOD	USER
SYSTEM	===LIBRARIAN	---DEVELOPMENT	PROCESS
LIBRARY	METHOD	PROBLEM	---WAY
---USEFUL	---ANALYSIS	---LARGE	METHOD
===HUMAN	GROUP	---THING	---DEVELOPMENT
SIMILARITY	---USEFUL	---WATER	PROBLEM
===RESEARCH	NEED	TOPIC	---LARGE
USER	---TIME	ACCESS	---THING
TOPIC	INFORMATION	PROCESS	---NEW
---WORLD	PROBLEM	===COLLECTION	ITEM
===PROBABLY	SIMILARITY	SIMILARITY	FREQUENCY
===GENERAL	---ORGANIZATION	---APPLY	---WATER
===REALLY	---WORLD	===IDEA	ACCESS
CASE	===KNOW	---WORK	---APPLY
GROUP	CASE	===CAPABILITY	CASE
SEARCH	===INDEX	---NEW	NEED
---TEXT	---TEXT	===POSSIBLE	===THINK
PROBLEM	ACCESS	CASE	===LANGUAGE
===DIFFICULT	SUBJECT	NEED	===COMPUTATIONAL
METHOD	---FILE	SUBJECT	===APPROACH
===NUMBER	===TECHNICAL	SEARCH	TOPIC
---ORGANIZATION	===MACHINE	ITEM	---TIME

Words occurring for only one of the drafts are preceded by double dashes (=), and those for any two are preceded by single dashes (-). One notes that the most similar pairs in the above lists are #1 and #2, having 28 words in common, and #3 and #4, with 29 words in common. Because #2 and #3 have only 23 words in common, one might draw the conclusion that the draft was changed more radically at that stage. This is a correct conclusion, as verified by the frequencies (in parentheses) of the top three words, which imply substantial lengthening of the draft at #3, and it is also evident that #3 was the draft in which the "water resources"

analogy was introduced; a less satisfactory "world is really flat" analogy, used in #1 and #2 (note presence of "world"), was dropped. It is also evident that the extended discussion of linguists' viewpoints was not brought in until the fourth draft.

Similarities are apparent in rank order that are also interesting to consider. The decision was made, however, not to use either the values of the frequencies or the rank positions of words representing individual text items as input to the classification programs to be discussed herein. There are both theoretical and practical reasons for not using such data: theoretically, from what might be called an "information theory viewpoint," far more information is embodied in the selection of the word out of the English vocabulary than its actual number of occurrences once an author decided to use it; practically, in the early stages of a research effort such as this, the input is kept as simple as the method allows, in order that the researcher can follow the workings or "mechanics" of his procedure. Even with unquantified word lists, as we shall see, things can get pretty complicated.

No two of the above lists have less than 50% commonality of word content; even the least similar drafts (#1 and #4, as might be expected) had 20 out of 36 words in common. The experience of this researcher has been that very few cases are found of any two word lists of that length having more than 50% common word content if the parent documents were independently generated; those few cases usually involve documents by the same author written within the same year, while he is likely to be reporting on the same sorts of activities. If one were to perform frequency counts on monthly progress reports, it is highly likely that most pairs of the frequent-word lists from the reports in consecutive months will have greater than 50% in common. As the time intervals between reports increase, the list similarities should steadily lessen. This very relationship was used as an "objective criterion" of classification accuracy in the experiments reported in 1965 (2).

Lists of the above length derived from papers by different authors working on quite similar research problems in the same field typically have from 12 to 16 words in common. But those from papers treating somewhat remote aspects of the same field (examples: papers in "documentation" dealing with topics as remote from each other as information centers and use of computers in searching English text) have from 5 to 10 words in common, and in this range--usually some of the common words are homographs. It has been generally found in numerous experiments like those reported in (2), many of which have not been published, that below "5 words in common" is a risk area; any list assigned to a category on the basis of having 3 or 4 words in common with one or more other lists will probably be misclassified. This leads to the ironic result that when categories are generated from, say, 100 topically close documents,

the subsequent document assignments via word similarity criteria will be far more free of errors than when 100 documents in widely scattered subjects are dealt with in the same way--in classification by human judgment it would seem that the reverse would be true.

This is not really ironic, but is at the heart of the explanation of the workability of classification according to word similarity: when one document has as many as a third* of its most frequently occurring content words in common with those of another document it is not only unlikely that the documents would be on different topics, it is even unlikely that the common words would have different meanings. By the time documents have become as topically close as the four drafts of Section I (#1 through #4, above), one can search for hours through the text before he finds enough instances of homography whose effect on the frequency count would change the degree of similarity of two of the lists. For example, a linguistic sleuth might plow through one of the drafts and, after 25 minutes, announce: "Aha! Here in this draft he uses the phrase 'in other words' twice, and in this other one doesn't use it at all. I can therefore subtract two from his frequency count of 'word,' since he is clearly using it in a different sense from 'words occurring in documents' in those two cases." Very clever, but this correction causes "word" to drop only two notches in rank, and the similarity is not altered. To reiterate, the actual numerical value of the frequency is of no significance; what matters is that "word" is used often enough to have a secure place on the list.

B. How a Classification Procedure Based on Word Similarity Operates

Before discussing the operation of our current classification procedure, we must make reference to a simpler, less efficient procedure, because through it we can show more directly how word similarities become involved in determining classification. One of the first classification runs made, using word similarities between lists like the above, employed the ward hierarchical grouping program (5,6), which operates in such a way that those lists having the greatest similarity are the first to be grouped; as less and less similar lists are incorporated in groups, the remaining ungrouped lists look more and more dissimilar, and finally the last few ungrouped lists have a minimum of words in common among themselves and with the other lists in the collection (note simplified illustration in Figure 1). This kind of automatic classification can be

*We can continue to think of frequent-word lists of 30 to 40 words in length for the remainder of this paper; the several exceptions will be evident.

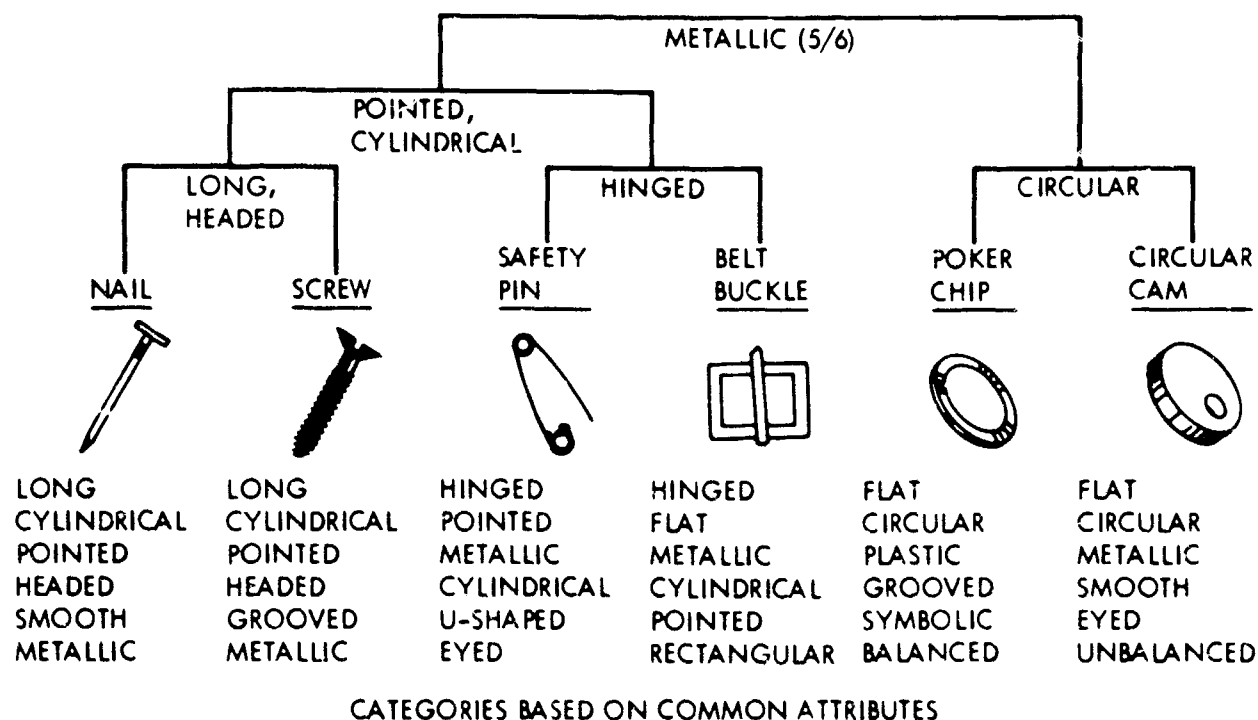


Figure 1. Derivation of Category Labels

This is a simplified illustration of the working of the Ward hierarchical grouping program. Each object shown is assigned six attribute labels. If these lists of six words each were used as input to the Ward program, each program cycle must combine any two entities (lists, list groups already formed) and only two, so that in the above case the following would happen:

Cycle #1. "Nail" and "screw" lists are combined, having five common attributes, a maximum for this input.

Cycle #2. "Safety pin" and "belt buckle" are combined, with four in common. If "safety pin" had been more similar to both "nail" and "screw," the program could have assigned it to that group instead.

Cycle #3. "Poker chip" and "cam," having low similarity, are paired. Notice that the program is running out of grouping possibilities.

Cycle #4. Since no ungrouped (unpaired) objects remain, this cycle must combine any two out of the three pairs, and chooses the two on the left. Note: If the average similarity of the four objects on the left had been high enough, this step could have occurred on Cycle #3. In fact, in one mode of the Ward program, using a different similarity function, it does.

Cycle #5. The only grouping possibility remaining is to pool everything. One version of the Ward program will assign labels to the groups it forms. Labels are selected from the member lists so that, ideally, words occurring on every list and only within the group would be chosen as labels. If this standard cannot be met, the Ward program will choose the closest approach to it, as in the case of the label "metallic," the word coming the nearest to aptly representing the entire group of six.

thought of in fanciful analogy to crystallization from a melt having many ingredients, as for example when sub-surface igneous rocks solidify. Just as the last few percent of material crystallizing out from the cooling magma is apt to be a heterogeneous mixture of slag and acidified brine, these last lists are often from documents that didn't belong in the collection, topically.

This early run of the Ward program was made in April 1964 on 100 36-word lists from documents about document retrieval and natural language processing. If the four lists shown above had been part of the input, the ones for drafts #3 and #4 would have been paired as the program's first group, followed immediately by those for drafts #1 and #2; five or six cycles later, all four would be grouped together because of their high average similarity to each other. The most similar lists in the 1964 run had 22 words in common, and were from the first and second half of a document on associative indexing. In fact, four out of seven of the first groups formed (and therefore composed of lists of highest similarity) were the pairings of lists from different portions of the same document. Thus, the word content from different parts of a document is almost as similar as the word content of rewritten drafts.

As the Ward program worked its way into groupings having lower similarity, it rapidly exhausted instances of recombining document fragments, and progressed into combining different documents by the same author and finally, documents by different authors on the same topic. The last remaining ungrouped documents (i.e., lists) were somewhat abstractly worded discussions of retrieval and semantics. Both were misclassified: the retrieval document was placed in a group with two parts of a document about "construction of a semantic code," and the one on semantics--which ought to have been placed there--was grouped with two documents about the process of doing research in information retrieval. Both had an average of five words in common with the other members of the groups to which they were assigned--the very threshold of classification reliability we described above.

C. Advantages of a Cluster Analysis Method Over the Ward Grouping Program

The classification method now being reported, which has superseded the Ward grouping method for large-scale use, has two basic advantages over the Ward program. As indicated in the opening of Section I, it is a "direct proportion" method, using computer time in proportion to the size of the collection of lists. Our version of the Ward program uses time in nearly cubed proportion to the number of lists.

The other basic advantage has to do with inherent classification accuracy rather than unit cost. This needs to be explained at some length. List-list comparisons are the basis of the Ward program's classifying; the majority--almost half--of the groupings are of one list with another, and

these pairs are subsequently augmented or combined to form larger groups. In the run described, 82 out of the 100 lists were paired first, and subsequently involved in larger groups. This meant that for the most of the documents a total pool of not more than 72 words (two 36-word lists) determined the initial categorization, and if any such categorization turns out to be not as good as could have been attained with more information brought to bear, the Ward program has no built-in means of correcting the situation.

The total message of the above run and all the others done in 1964 was that classification accuracy improves when the amount of information involved in determining the classification is made greater (2). Even the Ward program is capable by every extrapolation of outperforming human judgment* most of the time for inputs where topically close documents are represented by as many as 36 words. (This is an enormous amount of information, in the information-theoretic sense; the selecting power of 36 English words of typical usage frequency is greater than would be required to choose one atom of iron out of the total annual steel production of the United States. Notice that this does not mean 36 uncorrelated English words; if words occurred on the lists with no interrelations among their probabilities of occurrence, the selectivity would be far greater than that needed to choose one photon out of the observable universe.)

When the Ward program does make a mistake, it is a good bit more atrocious than an error in human judgment might be. All the selection power inherent in 36 words is not much good when it turns out that there is little to select from. Though human judgment might not be reliable (3), its departures from the norm are seldom outright mistakes because the human brain can bring to bear a huge amount of information to determine every act of categorization. For miscellaneous or odd-ball documents, the Ward program is relatively helpless; gigantic though the information content of 100 36-word lists is, it may omit the specific kind of information needed to categorize some of the more topically unusual lists.

The second basic advantage of the "direct proportion" method, as we can now rephrase it in the light of the above discussion, springs from its very ability to deal efficiently with large collections. This ability permits it to begin its operation with an information reservoir having

*"Outperforming human judgment" means both achieving a consistency of automatic classification (in the face of variations in input and method) that is less deviant than judgmental classification from person to person and avoiding obvious errors, as a human usually can. This is why "topically close" is underlined above, in that the Ward program makes many obvious errors in categorizing only the topically remote documents.

topical universality sufficient to provide appropriate categorizing information for all member information items. It can start "from the top down," whereas the Ward program, forming the small groups first and by nature confined to modest total inputs, can of itself only work from the bottom up. In Section III we shall see what truly enormous amount of information can be brought into play even in a small collection having a few hundred lists, when the "top down" route is followed.

D. Topical Profiles

It was told several paragraphs ago that most of the Ward program's categorizations in its operation on the 100 "retrieval" lists seldom are based on a total pool of more than 72 words. Our implementation of the "direct proportion method" is called ALCAPP (Automatic List Classification and Profile Production). It generates and uses as classifying agents very long lists of words called "profiles"; each profile may contain all the information of from a dozen to hundreds of individual lists. For collections greater than 100, no list ever has to be exposed to the typical information-lean situation of the Ward procedure, since only profiles determine categorization at the top of the hierarchy, and a lower limit of profile information content is easily maintainable.

It is necessary, before describing ALCAPP more specifically, to find some way of making profiles more "imaginable" in the hugeness and topical specificity of their information content. To harness the reader's intuition, suppose we were to arbitrarily divide the familiar public library into six broad topical areas:

1. Fiction, including children's books
2. Philosophy, religion, and history
3. Social and political sciences, including education and law
4. Economics, commerce and industry, human ecology, operations research and system science
5. Natural (physical and biological) sciences, including applied sciences and mathematics
6. Music, art, literature, dramatics, sports and entertainment, and travel.

If the books in each area were to be keypunched and word-counted to produce a list of frequent content words for each book, we could then pool all the lists in each area to form a profile of words common to the fields making up the area. If we arranged the words of a profile in order of the number of lists containing each word, with the words occurring most often at the top, we could see at a glance which content words were most typical of that area. But this may not be the most

satisfactory thing to do; consider for example what the results would be for, say, 2500 books in the fiction department. The top portion of the profile would look something like:

<u>Word</u>	<u>Number of Occurrences (cannot exceed 2500)</u>
SAID	2478
LOOKED	2315
TOOK	2275
REPLIED	2239
LAST	2213
RIGHT	2194
TODAY	2188
ASKED	2162
GOOD	2147
CAME	2144

These are surely marginal content words, but such a situation would exist not only for fiction, but for any broad subject area. However, suppose one sweeps his eye about a tenth of the way down the profile, what will it look like? For one thing, as a consequence of the well-known Zipf Law* (7), which asserts that there is a tendency for the number signifying a word's rank (in frequency) and the number for its frequency to be inversely proportional, frequencies reduce to quite low values even as soon as one-tenth of the way down on the profile:

<u>Word</u>	<u>Number of Occurrences</u>
TED	48
BIRD	47
CONTRACT	47
FINISH	47
LAME	47
SOFTLY	47
CLINT	46
EAVESDROP	46
KING	46
PATIENT	46

We now ask a question: which of these words will tend to be characteristic of the fiction section? In other words, even though they are not frequent, do some of the above ten words occur on this profile and nowhere else?

* In many cases of text, this relation is not strictly enough followed to be called a "law"; in the case of profiles the slope departs substantially from inverse proportionality at the top.

There are two obvious examples for which the answer is yes, "Ted" and "Clint," that we would not expect to be frequent in any nonfiction book. In other cases the answer is very likely no, such as "bird" and "patient" (which are probably also in the profile for natural and applied science), "king" (in history), and "contract" (in economics and possibly entertainment and sports). Thus if a measure of concentration of words on the fiction profile were to be used, and all words occurring too frequently on other profiles were rejected from fiction's profile, we would be left with a very amusing collection of words, consisting mainly of common first names, past-tense verbs, and adverbs (what a source of information for those who invent "Tom Swifties"!). Below are shown the probable results of a word-list and profile analysis for the other five divisions of the library:

Religion-

history-
philosophySocio-political
science and lawEconomicsNatural scienceArts and
sports

1) Probable highest ranking words, without regard for occurrence on other profiles

YEAR	ACT	SELL	SECOND	RECORD
LAND	LEARN	RATE	FIELD	PIECE
WAR	CONDITION	SHARE	POINT	PERFORMANCE
STATE	RELATION	PRICE	NUMBER	SCORE
ORDER	SUCCESS	COMPANY	TYPE	SEASON
WORLD	GROUP	TRADE	FORM	TOP
DEATH	HEAD	STOCK	PART	TOUR
GOVERNMENT	CASE	INTEREST	LEVEL	GREAT
POWER	CONTROL	PRODUCTION	BODY	WORK
MAN	MOVE	VALUE	MATERIAL	FORM

2) Words probably concentrated largely on one profile, but low in frequency

ARCHBISHOP	SAMOAN	AMORTIZE	ORBITAL	STANZA
ALAMO	TORT	DEBENTURE	FORCEPS	MOTET
TORQUEMADA	SENSORIMOTOR	NONDOLLAR	NEMATODE	LANDY
LEVANTINE	SOCIETAL	SEMITRAILER	ATP	ETCHER
VISIGOTHS	PROFESSORIAL	KEYNESIAN	TRIASSIC	MOTIF

3) Words probably both moderately concentrated on profile and fairly high-ranking

CENTURY	LEADERSHIP	COMMODITY	PARTICLE	EMMY
CONQUEST	MENTAL	VALUATION	SINE	GALLERY
SAINT	GUILTY	RESOURCES	ACID	PERFORMER
BORN	PROPAGANDA	SHIPMENT	TEMPERATURE	REHEARSAL
TREATY	MINORITY	STERLING	ABRASION	PLAYER
FEUDAL	CHILD	MERGER	BEAM	CHAMPIONSHIP
SETTLERS	PREJUDICE	CREDIT	CENTIMETER	AMATEUR
IMMORTALITY	COOPERATION	RETAIL	NUCLEUS	SPORTSMAN
FLEET	PARANOID	PACKAGING	ROTATION	OPUS
FRANCO	AGGRESSION	SPECULATION	FLUID	EXHIBIT

Words of type 2 are of the greatest value in identifying material belonging in each of the areas, because in addition to not being prominent on any but one of the profiles, they also occur often enough to have a high probability of being present on a word list belonging to that area. For example, it would be a rare article in experimental nuclear physics that did not contain the word "particle" on its frequent word list, and yet it is most unlikely that "particle" will occur more than a few times on any other profile. In Section III we will see the output of one of ALCAPP's routines that actually selects words like "particle" with pronounced affinities for only one profile.

Notice that the occurrence of a word, in order to be at a high rank on any profile formed from material as topically broad as "natural and applied science," has to have contributions from numerous different fields. Thus, "abrasion" would be: in medicine, a kind of injury, or a technique in plastic surgery; in mechanical engineering, a form of wear; in geology, an erosion process energized by wind and moving water; and in industrial technology, just another way of saying "sandblasting." (Observe how little need there is for a word like "abrasion" in the other four broad areas.) As another example, "acid" is a word that occurs often in medicine (certain drugs and stomach acid), genetics and virology (nucleic acid), agriculture (acidity characteristic of tropical soils), biochemistry, chemistry, and--probably most frequently of all--the domain of the chemical engineers (if the books of the latter do not use the most tokens of "acid," it is certain that their processes use the most tons of acid).

E. Iterative Classification Using Profiles

The use of topically specific words in automatic classification has been studied for several years by Williams (8). However, he begins with a priori categories (such as those we have arbitrarily chosen in the public library example), and uses the unique words to classify incoming accessions. As pointed out both in Sections I and II, this is not intrinsically a fully automatic process, even though Williams probably has made his version of it well on the automatic side of "semiautomatic."

Williams also regards the a priori categories as givens, not subject to improvement apart from human judgment. Suppose that we adopt the contrasting viewpoint of accepting a priori categories only as starting points on the road to better categorization. How would we attain improvement and what would it consist of? Williams uses his topically unique words only on accessions, which seems like a reasonable thing to do. Let us, however, do something which at first sight seems positively unreasonable: (1) use the entire profiles, not just the subset of unique words, as information to decide categorization, and (2) reclassify everything in the library, using the profiles.

There are, then, two corresponding questions:

1. Is it safe to use words that occur prominently on more than one profile? A word like "form," for example, might appear on all profiles, leading to unpredictable results in categorizing whatever lists contain that word.
2. Why should the five areas be better after reclassification than before?

First, we take up the question of using the whole profile. Notice, now, that the type-one words for the five areas are not without some uniqueness. Though Torquemada and the Visigoths are absolutely and exclusively on that profile, the word "year" ranks highest on the profile, and is not found among the top ten in rank on the other profiles. We may have to look as far down as rank 100 on the other profiles to find it. This differential in occurrence is capable of adding to the information available in determining classification; as we have concluded earlier, "the more, the better" when it comes to information brought to bear on classification. In the imaginary library situation above, there is enough information in each profile that perhaps we can afford to throw some away--and we might even want to for the sake of computational efficiency; for profiles generated from a smaller number of lists, though, it might be more advantageous to use all the information.

The key to avoiding the seeming ambiguity of using words present on many or all of the profiles to determine assignment to class is in weighted scoring. We have much more information available when we match a document's word list to a profile based on how high the list's words are ranked on a profile, rather than based merely on those words that occur uniquely on a profile. Height in rank can be used to score a word list by adding up suitable rank indicators, such as in Figure 2, where "rank value," i.e., some constant minus the rank, is shown in use.

A second consideration is that, rightly or wrongly, we are about to generate a new six-way breakdown for the library, and therefore can expect to see changes in the family of unique words; these changes may not occur if we give too much prominence to the unique words in determining classification. Obviously, the same books whose lists contributed to a word's uniqueness will reappear in the same area, on being reclassified, because the lists are not likely to contain any words that are unique on other profiles. Since profiles are generated from exclusively categorized lists, it is not possible for two words that occur together on many lists to show up as "unique" on different profiles; only one of the profiles can have both such words unique. If we take the position of not accepting the a priori classification as anything special, then we might as well not regard unique words as special either.

(Content of each group of lists is inventoried and listed.)

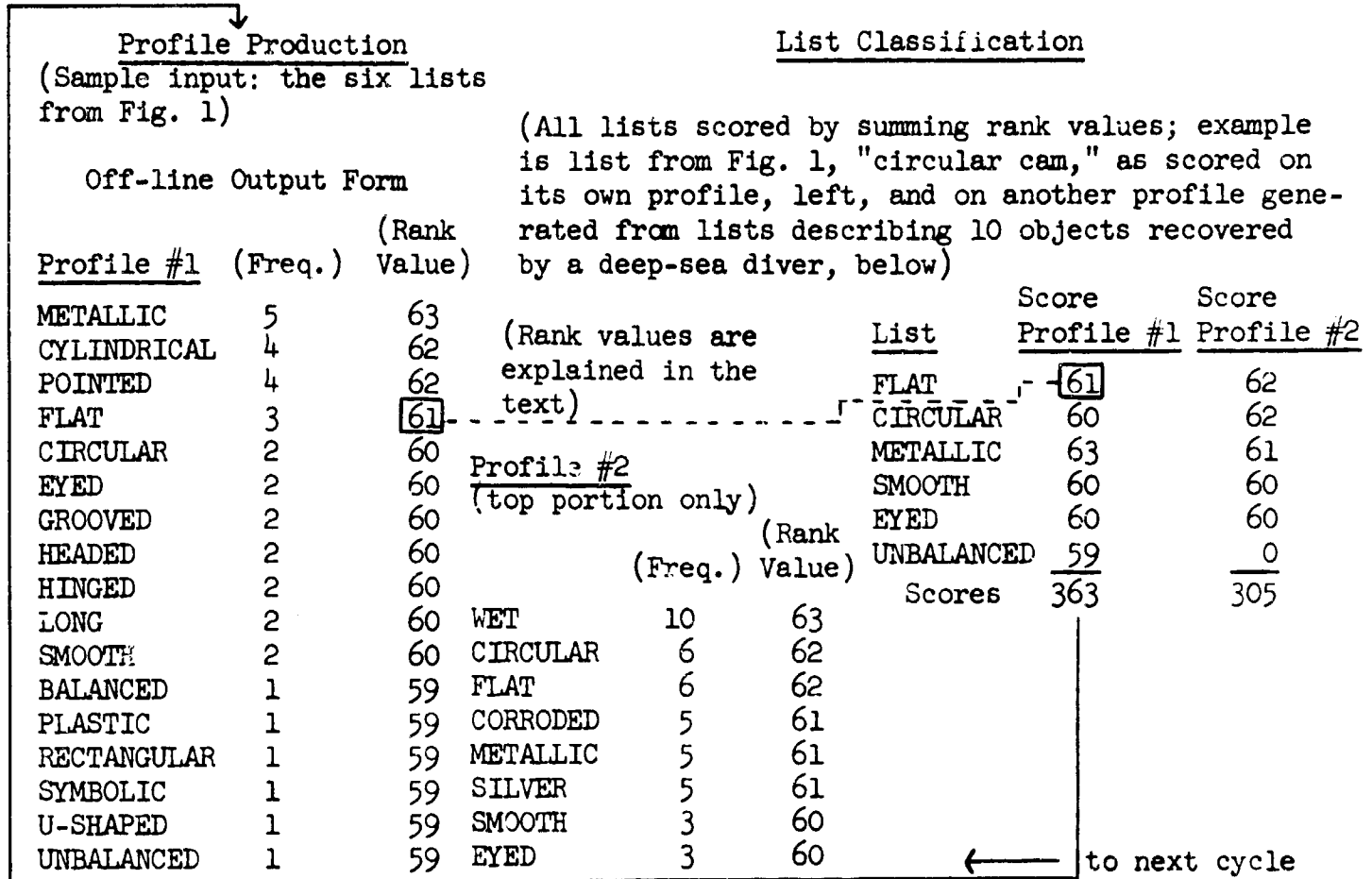


Figure 2. One Cycle of the Iterative Classification Process (showing the major components of profile generation and list assignment)

Explanation: A typical cycle of the iterative process begins at left with profile generation. The groups, such as the one from Fig. 1 used as an example, are formed before the first cycle either arbitrarily or based on a crude sorting mechanism; the arbitrary groups may be a subset of the whole list collection. After the first cycle, groups are determined by reassignment based on scoring each list with respect to each profile, as shown at right. The number of profiles equals the number of designated groups and ordinarily remains constant from cycle to cycle.

The cycle starts at left with the list groups as input, and for each group an inventory is made and used as a "profile" to represent that group; the frequency-ordered form shown here occurs only on printout; in storage the profile is in the form of a glossary, alphabetized for efficient lookup in later list-profile comparisons. List membership in groups is exclusive, so that one list can contribute to only one profile generated in a given cycle. At right, lists are reclassified on the basis of highest score, computed as shown; it is thus possible for lists to change profiles on the next cycle. Cycles are continued until "convergence," i.e., until no further changes of assignment of lists occur.

The next question is why we should expect reclassification to improve things. Here again there are two considerations. The simplest of these is that there are many debates and doubttings today about whether one pattern of topical distribution is better than another. There are boundaries between fields that are fuzzy (for example, do we want all material on law in one place, or do we want corporation and antitrust law under economics, and perhaps patent law under applied science?). Such fuzziness is on the increase with today's trend toward interdisciplinary relationships. Therefore, it is conceivable that reclassification might produce some appropriate changes and reveal unsuspected topical relationships.

A more basic consideration is that we are discussing classification technique applicable to document collections for which no a priori classification scheme is available; and, not to forget the statements in the beginning of Section I, we might even want to extend automatic classification to organizing items other than documents. If no judgmental classification scheme is used, then there are iterative techniques that can begin with just rough groupings by similarity and, by well-understood statistical criteria, can improve with each cycle like that shown in Figure 2 toward tighter or more internally similar groups. Not many cycles of reclassification and profile generation should pass before we attain the most internally similar or homogeneous groupings of lists. Section III will show that this does hold, even for a small collection of lists, where profiles may be generated from as few as 50 or 60 36-word lists.

Theoretically, in the above imaginary public library situation, the profiles would contain so much information that only about three cycles of reclassification and regeneration would be needed to reach a point at which no further changes take place in either the category assignments or the profiles. This stabilizing tendency is called "convergence." The convergence process can be recognized by the characteristic that at its greatest efficiency the number of items reclassified is at a maximum on the first reclassification cycle. In subsequent cycles the number reclassified should diminish quickly, if high-similarity clusters really exist in the collection of items.

Forgy (9) at UCLA has attained rapid convergence of this kind using medical data sets in grouping patients according to similarity in certain physiological attributes. The data are derived from medical measurements, so that each patient is characterized by a group of numbers in a fixed format, rather than by a simple list of words. The fact that his basic data are quantitative, and that each number represents a specific kind of measurement, leads to a very different situation for Forgy. For one thing he can work in terms of a spatial analogy, hoping to establish clusters in an n-dimensional space, with minimum sum of

squares, or "Euclidean distance," criterion of cluster membership; in this situation, both the concept and the practice of cluster-finding are simpler than is the case for our initially nonquantitative and unformatted word data. However, as is described in the next section, we are able to reach convergence from a "sloppy" starting distribution almost as rapidly as Forgy does; as of this writing, though, we still have some troubles in starting from a random or indifferent distribution.

The "direct proportionality" feature of the profile generation and reclassification process (which we have earlier denoted as ALCAPP) is feasible because exhaustive list-list comparisons are not involved. The Ward grouping program, discussed earlier, must begin by determining the similarity of each list to every other list; this must lead to at least a $T = N(N-1)/2$ relationship between computer time consumed and total number of items, N . Thus, if exhaustive list-list comparisons are required, there is no way to eliminate the n -squared factor in the proportionality. When total cost of classification increases as the square of collection size, then per-document cost increases linearly; this is easily visualized when one considers that "exhaustive comparison" means that a given document must undergo ten times as many comparisons in a collection of a million as in a collection of 100,000.

The actual intrinsic relationship between computer time and number and items is $T = N \log N$ for ALCAPP.* This is slightly greater than direct proportionality, but if the logarithmic base is large, the increment between $T = N$ and $T = N \log N$ may not add unreasonably to costs as n increases. The logarithmic base in principle can be made so large that the effect on costs of the $\log N$ factor is nil, and the value of the base is governed by the number of profiles one uses and is equal to this number if it is held constant at all stages of operations.

Suppose, for example, that one always sticks to 30 profiles (we used six in the foregoing public library example); then the logarithmic base is merely 30. What does this mean in operational terms? Let's further suppose that we begin with a collection of 30 million** documents, and use the following procedure:

*The constant of proportionality is omitted, to focus attention on the T-N relationship; the value of this constant is dependent on (a) speed of the computer, (b) actions of the program, and (c) the number of words per list.

**This is well beyond the capacity of ALCAPP in its existing form; the ALCAPP principle, however, does not require all input to be in memory during processing, and our version could be designed to handle an unlimited size of input.

1. Derive the word lists
2. Make a crude initial sort into 30 groups (crude sorting procedures are simple to design; good ones are not)
3. Generate the corresponding 30 profiles
4. Reclassify and regenerate profiles until convergence is attained.

At this point we have 30 groups of documents averaging a million each (and it is feasible to control the dispersion in group size). For further breakdown we simply repeat steps 3-4. The second breakdown leads to groups averaging 30,000 documents each. Third and fourth breakdowns are needed to obtain groups small enough to entail efficient use of procedure such as the Ward grouping program.

Note that each repetition of steps 3 through 4--each convergence to a 30-fold breakdown--uses approximately the same amount of computer time. At each successive breakdown one is in the situation of having 30 million lists, each of which must be compared to 30 profiles y times, where y , it is hoped, does not exceed four. (Note also, by the way, how this would contrast with comparing 30 million lists to each other: 435 trillion comparisons.) If y averages three cycles, one list must undergo, on the average, only 360 comparisons before reaching the most detailed categorization. If one works with 100 profiles rather than 30, one needs only three repetitions of steps 3-4 to reach the same level of detail, but this is more than offset by the need to make more profile-list comparisons. Therefore, contrary to what first seemed the case, economy arguments weigh in favor of a small logarithmic base; though this makes the $N \log N$ curve steeper, it reduces the constant of proportionality of the curve as a whole, enough to decrease costs all along the curve. For the collection of 30 million we are considering there is some optimum value of the logarithmic base that minimizes costs; since this value would probably depend on operational factors as well as on the sheer mathematics, there isn't much point in computing it--but a good guess would be a logarithmic base of about 7 or 8.

The more methodical reader might realize that a profile-list comparison should take much longer than a list-list comparison, thereby demolishing our cost comparisons. Actually, if the profile is in the form of an alphabetized dictionary the comparison time is quite small, because there is no need to inspect every word in a long, long profile; programmed dictionary lookup has been for years a highly efficient* process. An additional burden, however, does occur in the profile-generation part of

* It is noted that this efficiency also helps in the list-list comparison in matrix generation in those processes where computer time $T = Kn(n-1)/2$, especially when the similarity matrix has many zero values; but though this reduces the constant K , it does not remove the n -squared factor.

the cycle, although this has little effect on the overall relationship between processing time, total number of lists, and number of profiles.

F. The Potential of Large-Scale Qualitative Cluster Analysis

While this document was in preparation, a significant paper on clustering techniques came to the attention of the author, thanks to that most useful publication Computing Reviews. Written by Ball (10) of SRI, the paper surveys and discusses dozens of cluster-seeking techniques used in a wide variety of research fields. What is especially interesting is that, of all the techniques discussed, that of Ball himself is most like our own. It begins with a fixed number of more or less arbitrary cluster centers (corresponding to ALCAPP's profiles) and proceeds iteratively.

A conclusion of his is quoted here because it sounds so much like the first two paragraphs of Section I:

"We feel that computer-oriented techniques that can quickly organize data in a way that allows rapid analysis...will profoundly affect experimental science. Starting with existing clustering techniques and using proposed peripheral computer programs, it will be possible for the experimental scientist to see on a display the data he is gathering as he gathers it. The potential value in such rapid feedback seems enormous when we think how rapidly we forget all of the details of an experimental situation. We at SRI consider ourselves to be working toward this eventuality which may have considerable effect on the world around us."

Ball confines his generalization to experimental science, but ALCAPP was designed to be used in retrieval situations, and might be useful to anyone who must find information in fields having hundreds of thousands of information items. Its usefulness in experimental science, however, may prove to be limited, since most scientists design experiments to yield quantitative data, and are in a position to employ more refined techniques of cluster analysis. Some sciences surely must be stuck with situations where quantitative data might be hard to come by; we can only wait and see whether this is a very prevalent condition.

We can be more certain, however, that there are many agencies and individuals who must perforce deal with great masses of nonquantitative data, and of course document collections are the most well-known such masses of data. But there are many things besides documents that exist in large numbers and that cannot now be adequately dealt with. It is a big world, and possibly many who must cope with its bigness have not stopped to notice how much the already-existing organization of things in the world helps them in coping with it. Houses have numbers and both the street names and the names of the residents can be looked up in indexes and directories. Stores and supermarkets have things arranged

in orderly fashion, and the larger categories are labeled with varying degrees of visibility. Newspapers have different kinds of news and advertisements bunched in certain sections. Such examples that are familiar to everyone are too numerous to mention.

There are equally familiar frustrations, in dealing with a big world, that would be greatly eased by additional kinds of organization not feasible without computers. How many have wanted a certain kind of house or apartment, who have visited agency after agency or have driven for miles without finding one to meet what they first thought was a pretty reasonable set of requirements? How many have needed a specialized fixture that hardware stores ought (it seemed) to carry, but where in fact the proprietors queried didn't carry it and didn't even know about it? So many goods and services are becoming available in current times that it becomes harder by the day for people to be aware of them all. This very fact may turn out to be an unrecognized throttle on modern market economies. Most advertising doesn't help, since it is usually designed to aid the seller, not the buyer.

Departments of government contain notable examples of people whose jobs require them to wrestle with sheer bigness. The perennial need for federal agencies to collect statistics is a reflection of this. Some of these agencies have already found how much computers can assist them in getting full use of data that in former times were collected faithfully, but itemwise stood a poor chance of ever being looked at. The Internal Revenue Service is a well-known example. Yet even with present increases in efficiency, permitting universal checking of returns, etc., the income tax people are still limited to formatted information and the retrieval mechanisms that go therewith. Each taxpayer corresponds to a "tax situation," many of which become quite complex; only the internal revenue investigators themselves would know how much they might benefit by being able to group the millions of tax situations by pattern.

Income tax is far from the best example of the need for organization by similarity; usually people who monitor individual cases, such as tax collectors, can get so much good out of a nominal improvement in their data processing, as is presently true of internal revenue, that they do not especially desire even more capability. Wide publicity of even a poor computer-use technique in an area like taxes can itself bring enormous benefits just from the increased "honesty" it elicits.

The people in government these days who are most bogged by quantity and complexity are not the investigators, but the planners. The ability to retrieve individual information items by index tag is of relatively little assistance to them, since their problem is in understanding data about the world en masse. When the amount of data they have access to represents possibly millions of observations, cases, or people, now-available

statistical analysis procedures must be of limited help, both because of the "square-cube" cost factors we've talked about and because statistical processing ordinarily deals with measurements or summary counts, and not qualitative or descriptive (in a verbal sense) data.

If we were to program an upgraded-capacity version of ALCAPP, what sort of applicability would it have in government-level planning, and what kinds of qualitative data would be analyzed? We can imagine some specific examples, each corresponding to a different mode of usage; in this way we can illustrate two kinds of versatility of the profile-and-list iterative classification method, i.e., the diversity of problem areas to which it could be applied and the variety of analytical tasks it could handle in a given problem area. First, some usage modes will be described in the abstract, after which we present an example, for each such mode, in its application to a real problem area:

1. Patterning a large number of items into broad groups for purposes of resource allocation. This can be termed the "what-to-invest-where analysis."
2. Prediction from recorded event patterns, by grouping a large number of items defined as "complete," i.e., including for each item both a list of possible contributory factors and a list of types of outcomes, and by subsequently matching "unfulfilled" items, having contributory factors but no outcomes, to profiles at the desired level of detail. This can be called "extrapolation-of-experience analysis."
3. Isolating qualitative factors that make the greatest contribution to clustering, as a means of deciding kinds of information to emphasize in gathering data about a given population. We can call this "sorting-the-attributes analysis."

A possible application will now be described to illustrate how each mode might be used. It is necessary to point out that this author has no expertise in any of the areas pictured. Readers who may have knowledge in such cases can only be advised that if the shoe fits, it can be worn, but if it doesn't, the disclaimer has just been made.

The purpose of the examples below is only to stretch imaginations, not to stretch techniques to fit problems:

1. What-to-invest-where analysis in agricultural planning. Many under-developed countries and even some advanced countries that must import food might benefit from analytical techniques yielding a country-wide portrait of soil/climate factors affecting agricultural productivity. The planning made possible by such a soil condition inventory would aim toward maximum exploitation of the land with minimum investment in

irrigation, fertilizers, mechanization, etc., that are expensive enough to inhibit rapid improvement.

For each local tract of farmland there are many conditions that do affect or could affect crop yields. Most of these would be unfeasible or pointless to measure. Trace elements, such as zinc or cobalt, are examples of components that must be present in tiny amounts exceeding some threshold for the nourishment of some crops and domestic animals, and whose effects in concentrations above the threshold are either nonexistent or hard to determine. Also, there are many trace elements and substances not yet known to be important, but whose effects--or harmful absence--might be revealed in a large-scale analysis.

Added to this are factors of soil history (previous crops, etc.), of environmental agents (insects, types of periodic extremes in weather or microclimate), and abilities of local farmers. There could well be great variations of these conditions from one square mile to the next, or from one acre to the next. Data analyses that fully account for local variations and irregularities might be of substantial value in an intensively cultivated country like Japan, where crops are often grown between the roadbeds of double-track railways.

A complete analysis of this nature would enable agricultural planners to match the details of the country's crop-yield potential to its gross requirements for feeding its population and supplying its overseas markets. The soil-condition patterns that turned up would also provide a sound basis for allocation of resources toward irrigation, hinterland transportation facilities, etc.

2. Extrapolation-of-experience analysis in health and medicine. Medical researchers have become increasingly conscious of the importance of unexpected correlations among health disorders and ambient conditions. Just recently in the news was described a discovery that, among a group of cancer victims, 80% had once undergone an appendectomy, while in a noncancer control group of similar characteristics only 25% were without appendix. Forgy's work (9) is one instance of the trend toward studying patterns in physiological variables that might have bearing on some problems in pathology. For each relevant measurable factor, however, there may be a dozen relevant or potentially relevant factors that are, as we previously termed the, "unfeasible or pointless to measure."

Medicare poses coordinate sets of problems and opportunities. There has been much discussion of the problem of millions of claimants competing for limited facilities, etc., with perhaps not fully satisfactory plans for allocation of the facilities. At the same time, what have been called "indigent" citizens may in some sense not be indigent after all, for claimants to Medicare are capable of providing data about their own health.

The complete store of such data for million of claimants can be of great value to the remainder of the population if it can be analyzed.

Much of the correlation data we have heard about relate one thing and another, e.g., smoking and lung cancer, smog and breathing impairment, stress and atherosclerosis, etc. Some suspect that these correlated conditions are merely different facets of a complex underlying condition that we can't get at presently. At any rate, the instances of statistically related conditions can't presently be seen in the context of complete information about the affected population.

Imagine, now, that each Medicare applicant in a large population of applicants supplied 100 selected kinds of information about himself, ranging from childhood health incidents (tonsils removed, chicken pox, falls from trees, etc.), through long-term exposure to certain environments or activities (smog, low humidity, prolonged standing, outdoor work in cold climates, eating ice cream regularly, etc.), and perhaps extending even to data on the health of parents or siblings.

The kinds of groupings that would result from large-scale classification would be both impossible for us now to imagine as well as highly revealing. The predictive possibilities in such a large store of data are unguessable, but patterns are so ubiquitous in medical data that enormous benefits could ensue. Suppose, for example, that a class of 3500 patients is found, and that they are grouped together on the basis of each one having at least 20 out of a total of 32 common attributes. It might also be true that 90% of this group develops arthritis in middle age, and that 15 attributes are always present among those developing this disease, and present usually 10 or more years before its onset. What if you or I or the fellow up the street were to realize that 14 out of 15 of these attributes were true in his case? Such predictive possibilities may spring from environmental and hereditary influences in combination that are too complex to follow by any known analytical method.

3. Sorting-the-attributes analysis in juvenile delinquency. Juvenile delinquency is a problem currently at least as worrisome as the possibility of a dreaded illness. It occurs in rich and middle-class families as well as in the poverty-ridden; here again, its occurrence in huge numbers is both a measure of the size of the problem and the size of the opportunity for large-scale analysis. In treating juvenile delinquency, furthermore, the problem of choosing remedies is a much more puzzling one in the individual case than is the problem of choosing medication for the ill. This circumstance makes it as important to gather data on children who pull out of what looks like the early stages of delinquency as it does to accumulate data on those who drift into the pre-criminal pattern.

The same sort of predictive analysis that we discussed for medicine is applicable here, for some studies have shown definite causal elements. Here, however, human behavior is involved rather than physiology, and the associated, very complex cultural matrix, immediate neighborhood conditions, upbringing and parental influences, and sheer number of different ways of being delinquent present so many possible cause-effect relationships, not to speak of the combinations thereof, that one is in danger of not recording critical information about some kinds of delinquency situations.

Choices of information to be included about each item or case can have a variety of unhealthy consequences in cluster analysis. Leaving out the information of greatest value is only the most obvious of these. Inconsistencies of criteria and of designation are almost as obvious, when one considers how inherently vague is a question like "Did your parents argue frequently?" A less obvious hazard is that inclusion of information that contributes strongly to clustering, but that is really not relevant to the purposes for analyzing the data, can cause relevant clustering tendencies to produce little or no actual clustering.

An example of this was seen in an experiment done by the author in classifying 50 or so police department robbery reports using the Ward grouping program (11). Fortunately, variables were arbitrarily assigned two degrees of pertinence, so that on each 36-word list describing a single instance of robbery the topmost 18 words were called PBR (probably relevant) descriptors and the remainder PSR (possibly relevant). The classification program could be operated using either all 36 descriptors or only the top 18.

Six subdivisions of the 50 lists stood out when only the top 18 words were used. Group 1 consisted largely of mugging types of robbery where the victim was a lone pedestrian, with degrees of violence extending from stabbing to simple purse-snatching, but with no use of firearms. Group 2 incorporated all hotel and motel robberies, along with candy stores, a beauty parlor, and other of what might be termed "ultra-small business." Group 3 held most of the "conversational ruse" approaches to store robbery, in which merchandise is bought or questions asked to put the proprietor off-guard. Group 4 incorporated the unexpected twinned elements of a Negro suspect and a motorist victim, four out of seven of whom were cab drivers. Group 5 was also twin-faceted, having in common large takes averaging \$200 to \$300 and a geographical focus in the Hollywood-Wilshire area. Group 6 was more heterogeneous than the others, with one strange common factor: a revolver (never an automatic or other type of gun) was used in all seven of the robberies.

What was apparent, however, in comparison of the 18-word and 36-word runs was a pronounced tendency toward grouping on a geographical basis in the

latter case. It showed that if too much geographical or locational (sidewalk, alley, doorway, etc.) information had been included, other useful bases of clustering would have been masked. In any event, there are ways of knowing which descriptors or classes of descriptors contribute most strongly to cluster formation on account of their tendency to co-occur; in the robbery report analysis (11), a method of assigning labels to the categories produced by the Ward program was involved, choosing as labels descriptors from the lists on the basis of their tendency to occur on lists within the category but not outside the category. The labeling occurred at all levels of the hierarchy, so that the resultant tree of labels showed at a glance whether geography, description of the suspect, type of store robbed, or some other class of attributes was prominent in bringing about the clustering. More powerful methods than this are undoubtedly possible, so that unwanted clustering tendencies can be readily anticipated and damped out.

It is not to be regarded as incidental that such interesting groupings came from such a small sample of attribute lists, involving a mere 50 cases of robbery and determined by information that--by the time it arrived in computer storage--was really "fourth hand," having the distortions and information losses from the reporting by the victim, the codification by the intervening patrolman or detective, and the selection of descriptors by the author. What will happen when classification of this kind is based on a very large reservoir of information, with perhaps millions of items? We can only guess at what the redundancy in our environments and in ourselves will turn out to reveal. But we need not be restricted to guessing, for ALCAPP is available to characterize and sketch out all the diverse yokels, damsels, and critters of Dogpatch.

III. THEORIES, FEASIBILITY TESTS, AND PROJECTIONS

Despite our knowledge that within the last two years several researchers (9,10) have demonstrated the workability of iterative cluster-seeking programs, in particular the attainment of unique convergences in as few as three or four iterations, we know of no one who has shown either theoretically or experimentally that data like ours, inherently difficult to express as a distribution of points in n-dimensional and Euclidean-type spaces, could be handled successfully in an iteratively converging process.

There are undoubtedly ways to make our sort of data conform to a spatial model--perhaps the "n" in n-dimensional would be as large as the number of glossary entries, with permitted values of zero and one on each dimension. There may be ways that are not as cumbersome as dealing with several thousand dimensions, some of which bring in frequencies as dimensional measures. However, there is no reason aside from "making the problem fit a known model" that justifies handling word lists or "qualitative attribute lists" in that fashion. Assumptions about orthogonality

of semantic ordinates or about metric properties are entirely unwarranted, though they have been adopted repeatedly by many choosing a statistical approach.

Among other things, well-established empirical observations, such as Zipf's Law (7), clash with the assumption of an n-dimensional model. For texts following Zipf's Law, it can be readily seen that a large part of the n-dimensional space could never be occupied and that the coordinates of each spatial point representing word frequencies in a text sample would relate to each other with almost vice-like dependency (only so many ordinates can have a value in this range, so many more in this range, etc.). In general, the n-dimensional framework is too elaborate a description for phenomena that can be described much more economically (Zipf's rank-frequency curve, as a simple but not-too-fitting example, is adequately expressed in just two dimensions).

A. The "Why" of Lists and Profiles

One might protest that if a Euclidean-space model is not adopted, "What else is there?" There is a good deal else, and the 16-man-year investment in document-collection analysis at System Development Corporation has produced numerous empirical discoveries that are transmutable into the groundwork for models.

The foremost example is the relationship between information per item and classification accuracy, experiments about which were documented (2,11) and in other cases jotted down in notebooks. The evidence is not only more overwhelming than it needs to be, but one might wonder why such a principle needs empirical justification at all, being deducible from the most elementary considerations of sampling. Suppose the principle is phrased as follows:

Wherever there is order or redundancy in a given universe of items, the nature of the ordering is more and more knowable as we have access to larger and larger amounts of information per item.

When stated this way, the principle becomes recognizable as being one of the basic conditions that permitted intelligent bipeds to develop science. In our preoccupation with the scientific method we often forget about the sort of pre-scientific observation that must have led to the beginnings of the more methodical procedures that today characterize science, and even lose sight of the fact that in some areas this sort of observation is still important.

Looking further into the relation between "amount of information per item" and accuracy of classification, we ask: "What constitutes information per item, and how is it measured?" Here we are helped by some of

the underlying ideas of information theory, those connected with probability. In deciphering a coded English message, learning that what we have assumed is the letter "q" is followed by the letter "u" provides us substantial information only if we're not sure about "q." If we are dead certain the letter is "q," discovery of "u" would provide almost no information, being as highly expectable as it is, but discovery of a space is highly informative, since very few words end with "q."

We noted in Section II that the fact of occurrence of a content word intrinsically carries a large amount of information. If the word is "and," it is not as much information as for most words, since "and" is a very frequent and highly probable word. If the word is "platypus," the information is quite large, since the probability of "platypus" is so low. In a library setting, probability is related to selectivity, and less probable index terms carry more information and are more selective in retrieval operations.

These two genuinely theoretical considerations, therefore, that classification accuracy and information per item are related, and that the selection of the word itself--not the frequency of occurrence--embodies the information we need, both point to long lists of words unaccompanied by frequencies as being the best simple way of representing a document for automatic classification. (There may be better ways that are more complex, but we must know why the added complexity improves them before we can see that they are "better.")

Since we are doing a kind of cluster analysis, we also require the counterpart of a cluster centroid, such as is computed in cluster analysis in n-dimensional Euclidean spaces. But how does one compute the centroid of a group of word lists? Having already concluded that the word content of a list--not the frequencies that, by exceeding a threshold, cause the words to be chosen for the list--conveys the information about a document, we tend to want our "centroid" as well as our lists to reflect the word content of an entire cluster. This gives rise to the profile, described in Section II.

A profile can be a simple list of words itself, reflecting simply the total semantic inventory--a glossary, if you wish--of a cluster. For profiles, however, an additional need appears: we want to know which words in the lists produced the greatest amount of inter-list similarity; this means which words occurred on the most lists. If ten lists were clustered and yielded a profile, then any word occurring on all ten lists would lead to 45 pairs having increased similarity over what they would have had without the word. If a word occurred only on two lists, it could affect the similarity only of that pair.

Our requirement of the profile is that it should lead to the production of clusters having maximum inter-list similarity, with minimum similarities

between lists in different clusters. We are accordingly much interested in identifying and using the words that produce the greatest effects in forming such clusters. Therefore, contrary to our policy of having only words and not in-document frequencies on the individual lists, we want the profiles to have in addition to the words themselves some numerical information about the similarity-contributing capabilities of the words.

One more empirical finding influenced the way in which the similarity-contributing power of words was actually made use of in ALCAPP. We found very early in our experience with profiles that "flatter" profiles produced more accurate classifications (by criteria that had been established in previous experiments); this means that if some function of frequency (number of lists containing a word) is used in scoring* list-candidates for assignment to a given profile, and hence to a given cluster, better results are obtained when the function does not vary a great deal from its average value. In terms of the frequencies of words on the profile (and ALCAPP generates profiles having the words ordered according to frequency, the most frequent at the top), this means that something like the cube root of the frequency or the log of the frequency would give better classification results than the frequency itself or some higher power.

This makes sense in terms of things we already knew about amount of information and classification accuracy. Cube roots or logs of frequency permit every word on the profile to be more evenly involved in the scoring, thus effectively bringing the total semantic content of the profile to bear as information determining classification. At the same time, those words contributing most to similarity are still allowed to have greater effects on scoring: if cube root is used, a word occurring on 64 lists will have a weight of "4" in scoring, and this is still 4 times as large as "1," the scoring weight of a word occurring only on one list. Thus the "flat profile" is a compromise between two opposing requirements: our insistence on maximizing the involvement of all the profile's words (i.e., increasing the information) and our need to have profiles reflect cluster-generating influences.

However, one additional practical requirement led to the use of an unusual "flat function" of frequency in ALCAPP: there was a need to equalize the scoring power of profiles generated from different-sized clusters.

* Scoring in ALCAPP is simply finding which words on a list are present on a profile and adding the frequencies (or functions thereof) for those words to give the score for the similarity of that list and that profile. Then for all profiles the highest score for that list determines to which profile and corresponding cluster the list is assigned. See Figure 2, Section II.

When one cluster has 100 lists and another one only 10, then the profile of the former has both higher frequencies (or functions thereof) and more words than the latter's profile, and will invariably outscore it. Thus, like snowflakes in aging snow, the big ones get bigger and the little ones are soon wiped out. In our view, no advantage to a cluster should come merely from its size.

A number known as the "rank value" was chosen to be used in scoring; rank value is defined as some constant minus the rank. The present version of ALCAPP has 64 as the value of the constant, so that the rank value for the most frequent word on the profile (rank 1) is 63, that for the next most frequent (rank 2) is 62, etc. Actually, 64 is only the upper value of the constant; one option of an ALCAPP user is to choose some lesser value of the constant if he prefers. The choice of a lesser value permits the words of higher frequency on the profile to be more prominent in scoring, if in some cases this is desirable. This user-chosen value, however, is the same for all profiles, to preserve the equality of scoring power from one profile to another.

The use of rank value does not completely heal inequalities in scoring power as a result of differing cluster sizes. Thinking again of the 10-list profile versus the 100-list profile, if a value of 64 is used for the rank-value constant, the lowest possible rank value for any word on the 10-list profile is 54. The reason for this is that tied values in frequency are designated as equal in rank, so that if 6 words have occurred 3 times (and the top and second rank words, respectively, 10 and 9 times), all 6 are given a rank of 3, and in this case a rank value of 61. As one goes lower on the profile, more and more words tie in rank, so that by this definition all the frequency-one words are of rank 10 and rank value 54.

So, if we happen to be dealing with 36-word lists, probably 200 or more words will be on the profile, all of which can contribute 54 or more to the score of a given list with respect to that profile. In contrast, the 100-list profile has enough changes in frequency to permit rank value to descend rapidly, lower on the profile; it is possible for rank value to reach zero, but usually by the time frequency-one words are reached on the profile, rank values have descended to about 20 or so. Now, the profile for a 100-list cluster contains far more words than that for the 10-list cluster, possibly more than 1000. It might well be that the scoring power of a 1000-word profile, most of whose words would contribute less than 30 to scoring, would have about the same overall scoring power as a 200-word profile all of whose words are able to contribute more than 50 to scoring, the greater number of words in the former case being offset by the smaller average rank values. But we would guess this to be a coincidence, and in practice it is.

Two controls are available to an ALCAPP user to bring greater equalization of scoring power in the situation he happens to be dealing with. One, as we already mentioned, is to use a constant less than 64 as upper limit for rank value. Another is the choice of a "floor" for rank value; one can prevent rank value from being stepped down, as ALCAPP ordinarily does as frequencies on a profile diminish, below some set value. Ordinarily this value is "1," as a simple means of guaranteeing that all words on a profile will be able to contribute to scoring. We suspect that there is a better normalizing function than rank value, and we are keeping our eyes peeled.

B. The quest for Convergence

If clusters of lists having maximum inter-list similarity really exist, the criterion of finding them with an iterative procedure such as ALCAPP is a simple one: one must be able to "converge" on them from any arbitrary starting distribution of lists. Convergence implies a gradual or asymptotic approach to that particular distribution of lists corresponding to "the" clusters having maximum internal similarity. When one can repeat the convergence to the same set of clusters from different starting distributions, and repeats it time after time, he has demonstrated both that there are clusters in the data and that he can find them.

This is what the author and the programmer, Don Blankenship, set out to attain in November of 1965. Several factors made our quest for convergence a prolonged and arduous one. (1) No guidelines were known to us on how to converge with our highly peculiar data, involving intrinsically noncontinuous and nonmetric functions; we did not even have a way of knowing at the outset that convergence was possible, aside from mere intuition. (2) Our programs, called PROFILE and MATCH, the predecessors of ALCAPP, were operating in an experimental time-sharing system; and as production programs took hours of operating time (but of course modest compute time) for just a few iterative cycles, frequent system failures and overload situations regularly smashed or delayed our operations. (3) Many different kinds of controls were available to us, those we actually programmed and those we could easily program if needed, and in the beginning our choice and manner of use of these controls was arbitrary and often wrong; sometimes a couple of days of data analysis involving ten or fifteen pounds of printouts was needed to inform us precisely what effects the controls were having. (4) PROFILE (which generated profiles) and MATCH (which scored lists against profiles) were originally written with a different procedure in mind than iterative-type classification; this made them less efficient, but until we had accumulated enough experience in using them iteratively we could not know how inefficient they actually were; when it dawned on us, the programs were recoded and designated ALCAPP--Automatic List Classification and Profile Production; the new system was from 3 to 10 times as fast as the PROFILE-MATCH complex, depending on how it was used.

Some of the problems we ran into, of course, were those of the profile-based method itself, which are the only ones relevant to our discussion here. These were, roughly in the order that we became aware of them:

1. Metastability. This is a premature convergence to a distribution of lists which could not be subsequently duplicated, even approximately. Metastable distributions occur whenever the upper and/or lower rank-value constants are set too high, or--for a given pair of rank-value constants--whenever a cluster of lists representing atypical subject matter becomes too small. Either way, the contribution of any list to the profile to be generated on the next iterative cycle is large enough to guarantee that it will always remain assigned* to that profile in subsequent scoring; in effect every one of its 36 words has a corresponding word on a profile (naturally, since its word content went into building the profile) that will contribute 36 rank-values to its total score for that profile. The obvious solution for us was to choose rank-value constants low enough to encourage lists to migrate freely from one profile to another, at least in the first two or three cycles.

2. "Sloshing." This oscillatory migration of numerous lists back-and-forth between profiles is something of an opposite extreme from metastability, and caused us much more actual difficulty because of the simple fact that in preventing quick convergence, "sloshing" chewed up large amounts of computer time and fantastic amounts of researcher time. Sloshing also had a number of causes, which complicated our analysis of the trouble. There was a "fast slosh" and a "slow slosh," and the fast slosh turned out to have two sources, one inherent in the profile method and the other a result of an oversight in programming. The three (at least three) types of sloshing are described separately:

- a. The "unpredictable fast slosh" was a result of a programming oversight that failed to allow for the effect of ties in frequency on the overall scoring power of a profile. This slosh was quite devastating to morale because it often seemed to strike when our process was almost on the point of convergence.

One notes that rank values are computed without regard for the values of the frequencies, and of course without regard for differences in frequency of words next door to each other in rank. It matters not whether the gap in frequency is 1, 5, or 20, the rank counter is always stepped by one as the rank-value computing routine works its way down the profile.

*The obvious extreme case is where only one list is assigned to a cluster; the corresponding profile on the next cycle would be the list itself, with all of its words tied for rank 1.

If a tie in frequency value occurs, especially a three- or four-way tie, the words are given the same rank value; normally, this is fine, except when a profile isn't changing much, as is true when it approaches convergence. If on one cycle four words had different frequencies, but on the following cycle--as the fortuitous result of slight changes in frequency--had the same frequency (assuming no other words are present in that frequency vicinity), the words that formerly caused the rank counter to step by four units now cause it to step only by one unit, because of the tie. As a result, every rank value below that over the entire length of the profile is three units greater than what it would be without the tie, causing words ordinarily having low rank-values to change drastically in the contribution to scoring.

This profile then causes its cluster to be deluged by newly assigned lists captured from other clusters whose profiles remain roughly constant in scoring power. This sudden shift, like an earthquake, is followed by a series of wild oscillations in cycles to follow that gradually damp out until the next three- or four-way tie occurs. As soon as the trouble was diagnosed, the needed change in the rank-value routine was simple, involving adjustments of rank values to compensate for effects of ties.

- b. The "regular fast slosh" was a consequence of the profile method itself, and was an oscillatory migration of some lists back and forth between certain pairs of profiles from cycle to cycle. The rank-value method of scoring lists for assignment to profiles was used to eliminate the effect of cluster size in the scoring power of the profiles generated from each cluster, in other words to keep the big clusters from getting bigger and the small ones from dying out. This led to an overcompensating effect such that small-cluster profiles tended to produce larger clusters on the next iteration, and big clusters lost membership.

This of course was the general idea in the beginning, because we wanted to encourage clusters to be not too variant in size; and it was thought that having the big clusters become smaller and vice versa would be a stabilizing situation. Not foreseen were the oscillations from big to small, cycle after cycle, which damped out with great reluctance. Our eventual finding of a method of damping this particular oscillation, as we shall see, was the key to a more general success in convergence. On our first acquaintance with this instability, however, too many other competing effects were masking it and preventing us from seeing how to cope with it.

- c. The "slow slosh" was particularly frustrating because its "half wave length" seemed to be from three to ten iterations long; its implications with respect to the possibility of rapid convergence were

naturally quite disturbing. Actually, though we had no means of seeing it at the time, it was a first cousin of "metastability." It represented a premature aggregation of lists not as tightly bound as the clusters in which most of these lists would eventually lodge. The aggregations, however, were just "metastable enough" to permit the reassignment of their lists to more appropriate clusters on a piecemeal basis that took many cycles. (Later in this section we'll see an instance of "slow slosh" in action when a feasibility demonstration is described.) As soon as the relationship to metastability was perceived, the solution was the same, to prevent the aggregations from becoming too solid in the first two or three cycles.

3. Ambivalent lists. In studying the "fast slosh" movements we noted that certain lists were forever on the move between profiles, others changed only occasionally, and about a third of the lists didn't budge from their assigned cluster because the profile in the next cycle always gave them a high score for that cluster. Close attention to the most ambivalent lists showed that the ratio between their highest score on a profile and the next-highest score was close to 1.00. Fast slosh occurred whenever changes in profile makeup caused several lists to have their second-highest scores increased to highest scores, thus causing a change in assignment of the several lists to the cluster for which they were formerly scored next-to-highest; this reassignment affected the makeup of the newly generated profile for that cluster so that several other lists had their highest and next-highest scores trade places on the following cycle, and the transfer of these lists was usually to the cluster from which the original several lists had migrated. The two groups of several lists each would continually swap assignments to clusters, leading to the observed oscillatory movement.

We did not realize all at once that these ambivalent lists were the true culprits preventing convergence. The above portrayal of sloshing is necessarily oversimplified for purposes of description, and if the reader were given a true description of what we observed in all its gruesome detail, he would be as confused as we researchers were.

C. The Attainment of Convergence and a Novel Demonstration. The first case of what looked like genuine convergence occurred on May 22--six full months after the beginning of our quest for convergence--and was attained by Don Blankenship, operating the freshly programmed ALCAPP. This improved version of the profile-based method, in addition to being fast enough to permit about eight cycles an hour, even under busy time-sharing system conditions, also contains two different ways of preventing ambivalent lists (lists with ratios close to 1 of the highest score to the next-highest score) from taking part in building profiles on the next cycle. This prohibition, it was expected, would damp out the effects of the minor oscillatory migrations on the composition of the profiles, and hence on their scoring power. This proved indeed to be the case.

This provision, however, takes care only of fast sloshing. To our good fortune, though, it happened that a variety of effects could be achieved simply by varying the "ratio threshold" that screens out the ambivalents when the threshold value is set close to 1.00. Other useful effects occur when this threshold is set to values remote from unity, so that only the lists with quite high preferences for one profile can participate in profile-building on the following cycle. This form of control leads directly to forming the tight "inner nuclei" that are the backbone of the truly stable clusters towards which ALCAPP should converge. The formation and identification of these nuclei, or "inner clusters," did in fact clear up all the other problems of the iterative method.

The formation of the inner clusters eliminates the slow slosh because it produces immediately the configuration that was being "slow-sloshed toward" from the prematurely formed, less coherent aggregates. It also is the key to rapid convergence, because the inner clusters--once formed--require the equivalent of dynamite to break them apart; the more vigorous maneuvers that can then be made, by changing rank-value constants and other control parameters, can prevent the minor incursions of metastability that put a brake on convergence.

There is only one remaining "unsolved" problem, which, under the new perspective, is difficult to think of as a real problem: that in lists derived from topically close material there seem always to be ambivalents. This however cannot be attributed to the method, but is inherent in the structure of the data; the same situation obtains in spatially clustered objects, where some cluster members are strung out close to the outlying portions of neighboring clusters. The main thing we expect of our procedure is to lock in, quickly and unerringly, on those subpopulations that unequivocally belong in the same clusters. This is what can now be done with ALCAPP.

This is the point in this discussion, now that the suspense has been built up, to "show the reader" that ALCAPP can indeed lock in on the same clusters from different and arbitrary starting distributions, in a small number of cycles, and that the final distributions (i.e., the list-membership of the clusters) will be the same except for a small number of ambivalent lists.

As an author who likes to be not merely scientific, but convincing as well, I have given considerable thought to the problem of "showing the reader" in such a way that a bare minimum has to be taken on faith. It is quite respectable to use a random number table to determine the list-numbers that make up my arbitrary starting distributions, but this has the disadvantage that the reader is not present to watch me look up the random numbers. He may not doubt my honesty, but there is always the chance that I might unwittingly make some subtle mistake. More important, "seeing is believing" is an oft-neglected cognitive truth.

So far as it is possible in a research document, there must be a way to personally involve the reader in the selection of the starting distributions, then to show him how the chips fall as a result.

Fortunately, choosing random numbers is only one of many possible ways of choosing an arbitrary beginning. What we really want is not necessarily random numbers, but indifferent numbers; the starting word-lists must be chosen in a way that cannot reflect any information derived from the similarity relations in the data. There are many ways to secure numbers having no connection at all with our data--take them from the license plates of passing cars, extract the last three digits of telephone numbers starting at the top of page 111 of the nearest directory, call up all one's friends and ask each to name a number from 1 to 419 (we are going to use 419 36-word lists in this demonstration), etc.; these are all indifferent number-selection processes, but none of them has the desired property that the reader can be present to see them selected.

There is a way, however, to have the reader present. Only one qualification must be made: the reader must not believe in "word magic." Any reader who believes, for example, that 20th century presidents must have double letters in their names or initials (except LBJ for reasons that are explained by word magicians at great length), or that it would be unfortunate to live on 13th street, or that to tell about how lucky one has been will bring an end to that person's luck, is attributing mysterious powers to certain combinations of words, letters, or other symbols; he is addicted to "word magic." Such a reader could not possibly be convinced that the process about to be used is really indifferent.

Immediately below will be given four lists of words. Each list was derived from one of four profiles at convergence, and the words were chosen from their corresponding profiles on the basis of high rank and strong preference for the profile (these words would be equivalent to those in the bottom group for the five nonfiction areas in the public library example given in Section II). Such words are selected at the user's option by the profile-generating part of ALCAPP, according to the scoring algorithm: $S = R^2 / \sigma R$, where R is rank value of the word on that profile and σR is the total of the rank values for that word on all the profiles (i.e., on four profiles in the case to be considered). As can be seen from inspection of this formula, a word present on all profiles at about the same rank would have only one-fourth (we assume four profiles from here on) as high a score as it would if it were present at that rank only on the one profile, because of the σR divisor. The R^2 numerator guarantees a better score, other things being equal, for words ranked higher on the profile. The lists below have the highest scoring words at the top, and are chopped off arbitrarily at the 12th highest scoring word:

<u>Profile #1</u>	<u>Profile #2</u>	<u>Profile #3</u>	<u>Profile #4</u>
PROBLEMS	SIGNAL	COOCCURRENCE	SERVICE
INDEXING	SIGN	TAG	SCIENTIST
DOCUMENTS	MEMORY	COUNT	STUDENT
TECHNIQUES	EQUATION	RESEARCHER	CATALOG
PROGRAMS	SWITCH	EMPIRICAL	ENGINEER
WORDS	CIRCUIT	MARON	AGENCY
TERMS	TRANSFORMATION	PERMUTATION	ECONOMY
SYSTEMS	PULSE	HOMOGRAPH	DISSEMINATION
PROCESSING	ZERO	STRONG	PROFESSION
LANGUAGES	POSSIBLE	CLUSTER	COMMUNITY
PARTS	SENSE	TOKEN	EDITOR
USING	PERMANENT	MARKER	SPECIALIZE

And now, for the delight of those who believe in it and for the edification of those who do not, we apply "word magic." From each of the above four lists we try to convert letter combinations into numbers that will correspond to lists in the collection used as input for ALCAPP. We use the following rules to do this:

1. Beginning with the top word on a list, find the ordinal number in the alphabet for the word's first and second letters. Subtract one from the first number and multiply it by 16; then add the second number. By this rule it is theoretically possible to generate 416 out of the 419 list numbers. The final three list numbers are omitted, but this should not affect the basic "indifference" of the selection rule.
2. Beginning with the list for profile #1, generate 10 numbers for each profile by using the first and second letters of the top 10 words (out of 12) according to rule 1. If a number turns out to be identical to one previously generated, proceed to the third and fourth letters in the word, or to the fifth and sixth if need be, etc., to generate a number not identical to a previous one. (Thus, for the word "programs" on the list for profile #1, the letters O and G had to be used so as not to duplicate the number generated from "problems.")
3. Duplicate numbers are prohibited from corresponding to adjacent profiles as well as to the same profile. If a duplicate is generated, apply the appropriate part of rule 2 to the word belonging to the higher numbered profile (the one farther to the right, above).
4. The numbers are directly usable as teletype input to the profile-generating portion of ALCAPP.

Using these rules, the following sets of numbers were generated (manually) for the profiles, arranged for convenience in numerical order:


<u>Profile #1</u>	<u>Profile #2</u>	<u>Profile #3</u>	<u>Profile #4</u>
63	41	44	7
142	81	47	33
177	110	77	57
227	197	127	67
231	255	193	78
258	261	245	205
285	297	277	230
309	311	305	291
313	322	308	293
367	405	336	324

How "indifferent," actually, are these number selections? The most direct inquiry into the situation is to inspect the distributions that resulted when the above profiles, from which came the "magic words" we just used, were used to reclassify the lists on the following cycle. It is quite easy for me to make this investigation because the numbers for the reclassified lists are printed out on the same strip of teletype paper that contains the magic words. I find that:

1. Scoring highest for profile #1 were the lists numbered 7, 33, 41, 47, 57, 63, 77, 78, and 81. Notice that "word magic" has treacherously placed four of these list numbers with profile #4.
2. For profile #2 were lists numbered 44, 67, 127, 261, 277, 285, 291, 293, 297, 324, 336, 367, and 405. These lists were scattered by "word magic" among all four profiles; however, keep an eye on 291 and 293, which--as it turns out--are going to make for ALCAPP not merely an indifferent starting distribution, but a distinctly unfavorable one.
3. For profile #3 were the lists 193, 197, 205, 227, 230, 231, 245, 255, 305, 308, and 309. "Word magic" has kindly permitted four of these lists to return to their favorite profile.
4. For profile #4 were the lists 110, 142, 177, 258, 311, 313, and 322. But for some strange reason, "word magic" put them all with profiles #1 and #2. Can anyone doubt how villainously indifferent this number selection procedure is?

Actually, our interest is only to generate the same four clusters of lists that arose from the profile from which we selected the magic words. We don't care if the profile numbers don't correspond, since the profile numbers are merely arbitrary labels. Can we predict what the new profile

numbers will be, assuming that we succeed in getting roughly the same clusters? A four-by-four table comparing the word-magic assignments and the assignments resulting from ALCAPP's use of the four profiles might help us:

		Profile #1	#2	#3	#4	← (These profile numbers are for the starting distribution)
(These are the original profile numbers)		#1	#2	#3	#4	
	#1	1	2	2	4	
	#2	2	3	4	4	
	#3	3	2	4	2	
	#4	4	3	-	-	

A number in this table shows how many lists legitimately classified under a given profile number were distributed by "word magic" to each profile number in our indifferent starting distribution. For example, the number 4 at the lower left means that four lists scoring highest and duly assigned to profile #4 by ALCAPP were reshuffled under the profile #1 heading by "word magic." This, it would appear, slightly biases things in favor of the cluster formerly associated with profile #4 being regenerated as profile #1. For the other three clusters the outlook is not so evident. Profile #4 for the starting distribution looks equally biased towards the regeneration of the clusters formerly with profiles #1 and #2, and as we shall see quite a few iterative cycles will be needed for profile #4 to "make up its mind." A random number table, by the way, will give the same kinds of biases, as the reader can demonstrate for himself in less than 15 minutes.

Note, however, that the profile #4 lists, though they come equally from the original profiles #1 and #2 (4 lists from each), are unequally counterbalanced on their profiles in the starting distribution. The effects that the four lists from #2 would have during the first iteration are offset by the presence of four lists from #2 also under the (new) profile #3 heading; no such counterbalancing is seen of the four lists from #1. Such considerations as this permit one to make the following prediction: If the clusters are regenerated, the one associated with old profile #1 will now be with profile #4; that with old profile #2 will still be with profile #2; the same will hold for profile #3; finally, in the most easily predictable case, the new profile #1 will inherit the cluster that used to be with #4.

A comment on the unusual nature of the old profile #1 is needed, referring back a couple of pages to the lists of words "most unique" for each profile. The observant reader probably noticed that all of the words given under profile #1 are either plurals or participles, which do not occur at all under the other profile headings. This is no coincidence, but is an interesting comment both on the brute power of the ALCAPP classifying algorithm

and on the wisdom of using consistent policies of dealing with suffixes in input derived from natural language.

It happens that about 20% of the 419 lists in our input were prepared as part of an experiment to determine the effects on classification of not normalizing suffixes. The results of that experiment were inconclusive, but this is beside the point; the point of interest is that eventually when our classification project needed a fairly large corpus of topically close material, we scraped together everything we had on hand, so that the unnormalized lists were thrown in with lists whose words were suffix-normalized, in accordance with our usual policy.

The unnormalized lists (86 of them) had on the average 10 words on each 36-word list having a suffixed variant--in 9 out of 10 such cases, a noun plural. From the viewpoint of the semantically ignorant word-matching routine of ALCAPP, the singular and plural forms of a noun are two different words. We could easily have fixed this defect of ALCAPP (there are plenty of "suffix splitting" programs around), but frankly the researcher who assembled the input was curious to see what would happen; he saw, and now we see. The plurals--which might as well have been foreign words as far as the program could tell--exercise decisive effects on the classification of those 86 lists; and, as we shall see, an approach toward convergence invariably results in all but two or three of the 86 winding up in the same cluster. Furthermore, when the actual values of the "uniqueness scores" (remembering that the scoring algorithm is $S = R^2/\sigma R$, already explained) are looked at, those for the 12 words most unique to profile #1 were all higher than any of the 36 unique words for the other three profiles; this means that the unique words are more frequent on that profile and contribute more heavily in scoring for list assignment.

The cluster having the lists with suffixed words is definitely the most cohesive, and in our experiments with the 419-list 4-profile situation has usually been the first cluster to form. It is useful for the reader to be aware, in following the iterative cycles to be described shortly, that only after the most cohesive cluster acquires most of its membership can the weaker, less cohesive clusters begin to take shape. As a corollary, any influence that interferes with the formation of the most cohesive cluster will tend to prevent the other clusters from assuming the form they would ordinarily take one.

In this light, the "indifferent" starting distribution that the word-magic procedure has selected is actually unfavorable, all because of two lists, numbers 291 and 293. First let us visualize in abstract terms the effect that these two lists will have. Each cluster ordinarily is composed of subclusters; this is especially the case when it comes to clusters based on document similarities. As was true for the most cohesive clusters, the most cohesive subclusters also form quite early; they can and do

gravitate to a profile not associated with the cluster to which--on a similarity basis--they really belong. As was pointed out earlier, lists 291 and 293, and the subcluster of which they are a part, ordinarily join the cluster of profile #2 (the one whose unique words are "signal," "sign," etc.), and have little similarity to the lists clustered with profile #1.

We are at the beginning of the first iteration. We have logged in on one of the teletypes plugged into the Q-32 time-shared computer, and have loaded ALCAPP and the 419 36-word lists; ALCAPP is rotating on one of the Q-32's five drums, ready to be used, and the 419 lists are in a disc file that we have labeled OCELOT (not an acronym). You, the reader, have taken a seat at the teletype and typed "GO." ALCAPP whirls into operation, asking for a command. You type "PROFILE"; this requests the profile-generating half of the program.

ALCAPP then asks for a label for the profile set it will generate, for the name of the distribution of lists to be used (remember that each profile is an inventory of the word content of a group or cluster of lists), and for instructions about kinds of output desired. This request types itself out automatically on the same teletype used for input; data as well as program requests can be given as teletype output, creating a convenient time-ordered record of the actions of the user and of the program.

You do not know a name of a distribution. Naturally not, since you are going to feed in via teletype the starting distribution chosen by "word magic." So in place of a name, you type "TTY," the code meaning distribution will come through teletype rather than from disc storage. ALCAPP then asks how many profiles you wish to build, and of course you type "4." Then teletype sputters "GROUP 1," and the reader knows he should enter the numbers of the ten lists for profile #1. This process is repeated three more, after which the teletype becomes mysteriously silent. You say, "What's happening?" The programmer standing nearby says, "It's thinking."

This answer, of course, is an oversimplification as well as an anthropomorphism. Since the computer is time-shared, many other programs are taking turns operating. Building these profiles takes several seconds of actual program operating time, but it must be done in units of up to 400 milliseconds, and between consecutive units as many as 20 other users might be serviced--though it usually doesn't require 400 milliseconds for each of them. At any rate, the several seconds of actual compute time will be stretched out to perhaps a half-minute of real time.

No sooner does the programmer finish saying, "It's thinking," than the teletype begins to chatter. It is spewing forth words, the "unique words" for the profiles ALCAPP just generated, and they are as follows:

<u>Profile #1</u>	<u>Profile #2</u>	<u>Profile #3</u>	<u>Profile #4</u>
EFFORT	TEST	FACTOR	PROJECT
THEORY	FILE	EXAMPLE	CODES
APPROACH	B	LEVEL	GENERAL
EXTRACT	TAPE	PARTS	CHECKING
EASY	SCHEME	AUTHOR	DESCRIPTOR
HANDLING	CURRENT	SIMULATE	FORM
METHODOLOGY	OBJECTIVE	FIELD	ROBBERY
MODEL	COLLECTION	INTERPRETATION	PARSING
PATTERN	COUNT	REPRESENTATION	SENTENCES
APPLICATION	SUM	SENTENCE	COMMON
PAPER	STUDY	ELEMENT	NATURAL
LITERATURE	SIMULATION	CERTAIN	SUSPECT
CARD	ARTICLE	BOX	PHASE

You, the reader, are distinctly puzzled, commenting, "But these words don't look anything like those for the clusters we're trying to duplicate." The nearby programmer advises patience, pointing out that these profiles after all were based not on clusters but on 10 indifferently selected lists. Word magic or not, these words just have to be different, if for no other reason than that only 40 lists are involved in building the profiles and not the full 419. Nevertheless, if the Q-32 Time-Sharing System were to fail at this point, you might well walk off in a huff with the indelible impression that cluster analysis really is a form of alchemy, as you've often suspected.

But the Time-Sharing System (TSS) continues to function, and ALCAPP awaits the next instruction. You comment, "What's with 'robbery' on this fourth profile?" The programmer, who has spent many hectic hours working with this group of lists, is depressingly familiar with them and their parent documents. He says, "These two lists here (pointing to the input lists numbered 291 and 293) were generated from different parts of the same document. There are nine such lists all together, and they all talk about using our programs to group robbery reports. Putting those lists in was Doyle's idea, not mine." The parent document, by the way, is the one discussed at the end of Section II (11). The nine lists make up the subcluster that will, for several uncomfortable cycles, camp on profile #4.

We can now use the profiles just generated to separate the entire 419 lists into four groups. These groups will not be anything like clusters, but they will have interesting properties indicative of clustering potential. You type "MATCH," indicating the second major ALCAPP function, the scoring and assignment of lists according to profile. Again information is requested by the program the name of the disc file containing our profiles, the name desired for the output file (the distribution of

lists), types of output wanted, and type of scoring wanted. You type in "1" for the latter, indicating that you want all 419 lists matched and distributed.

Seconds later ALCAPP requests scoring parameters for the matching function. "What do I do now?" you ask. "Anything," the programmer helpfully suggests, then: "The first two numbers are your upper and lower limits for rank value. At this stage in the procedure the values you use aren't too critical." The last time you had a choice of a parameter value, you typed "4" for four profiles, so you nonchalantly type it again as an upper limit for rank value, and type "0" as a lower limit. We wait expectantly as the matching function goes into the TSS production stack, a special queueing arrangement for programs requiring more than 3 or 4 seconds of operating time between interactions at teletype. This phase requires a minimum of 18 seconds of compute time for the 419 lists, and the profile-generation phase requires 16, giving a total of 34 seconds per iterative cycle.

Teletype informs us that the lists have been distributed: profile #1, 7 lists; #2, 185 lists; #3, 221 lists; and #4, only 6 lists. "Four wasn't a very good number," you observe, "but why not?" The programmer points out that only a few steps in rank value are needed to reduce it to zero from such a small upper limit, and this is likely to happen unevenly on different profiles. In this case it happened so unevenly that three times as many words took part in the scoring on profile #3 as on profile #1. Since the profiles have been generated from unclustered lists, it is a wonder that profile #4 has any lists assigned to it at all.

In the early stages of this procedure, before clustering tendencies have had a chance to show themselves, it pays to keep the groups from which profiles are generated at about the same size, as the most direct way to insure equality of scoring power; once the cluster nuclei have formed, the rank-value scoring function will be adequate to check the effects of group-size variations. The programmer therefore suggests the use of mode 3 of the matching function on the next try; this will select for each profile some number n of lists having the highest ratio between the highest and next highest scores, as explained earlier in this section.

You use mode 3 with $n = 20$ and a top rank value of 6. This time the results look satisfactory enough, and we are ready to begin the second iteration. Once again you request the profile-generating function, and this time the following lists of unique words are printed out:

<u>Profile #1</u>	<u>Profile #2</u>	<u>Profile #3</u>	<u>Profile #4</u>
SEARCH	TEST	EXAMPLE	PROJECT
YEAR	CURRENT	ELEMENT	ROBBERY
APPROACH	NUMBER	ASSOCIATION	DESCRIPTOR
REQUEST	TAPE	SENTENCE	GENERAL
TAG	ERROR	TOPIC	CODES
BOOK	LENGTH	REPRESENTATION	SUSPECT
INTELLIGENCE	LINEAR	RULE	COMMON
PRECISION	BIT	RELATIONSHIP	VICTIM
SCIENTIST	IBM	PHRASE	NATURAL
PAST	COVERAGE	DISTINCTION	BETTER
EARLY	EXTENT	CLUSTER	SOUTH
CHANCE	ENTRIES	FACTOR	EARLIER

You look hard and find three of the unique words that were on the profiles we're trying to duplicate. Two of these, "tag" and "cluster," are from original profile #3, but alas in the present instance they are on different profiles; you comment that this is not a very auspicious indication. "Don't get excited," says the programmer, "we still are only using a small part of the input to generate profiles."

Mode 3 is again used for matching, this time with $n = 45$; this means a total of 180 lists will be used in the next round of profile building. Since we are curious as to which lists are scoring highest on which profiles, we indicate each group of 45 list numbers to be printed out on teletype. Inspection of these numbers, however, gives neither the reader nor the programmer cause for cheer. The 86 lists containing the suffixed words, which we are following as a result of knowing that this is the tightest cluster and also the key to the whole show, are far from concentrated on one profile. We predicted profile #4 for this cluster, but less than half of the printed-out list numbers from 1 to 86 are present there. The actual numbers of lists from that group for each profile are: #1, 12; #2, 10; #3, 4; and #4, 24. The remainder of the 86 were excluded because of low ratios; we have our fingers crossed and hope that the 24 lists on profile #4 is enough of a preponderance to attract the others.

Another thing is evident that disturbs the programmer more than it does you, the reader, because the programmer hasn't yet seen this happen with ALCAPP, though it used to happen routinely with the less effective programs before ALCAPP. He sees that lists 287 through 295 are all present to take part in building profile #4 the next cycle--the nine lists from the document about robbery reports, every last one of them! That a thing like this should happen while he is demonstrating the program to a skeptical reader is just unthinkable. "Word magic," he is heard to mutter. He knows from previous runs that those lists don't belong with the other lists that are gravitating to that profile. He considers

starting over again with a better starting distribution, but it is too late; too many embarrassing questions will be asked by the reader.

Actually, we should consider ourselves lucky that the starting distribution was unfavorable, because if our classification algorithm is any good it should pull us out of this, and if we can pull out we'll have the reader there watching. The third iteration begins, and the most unique words are again printed out for each profile. Things are looking a little better: six unique words from the original profiles are present (it was three the previous cycle), including the words "indexing" and "languages," both unique for profile #4.

We undertake the third use of mode 3 of the matching function, increasing n to 75. Studying the list numbers, we see that 70% of those from 1 to 86 are with profile #4, as compared to 48% the previous cycle. Unfortunately, all of the lists from the robbery report document are still present.

Cycle 4 begins, and a short time later we see that 15 of the unique words from the original profiles are also rated unique in this run; in just two cycles the number of such words has increased from 3 to 15. Furthermore, all 15 are distributed properly, and in a manner that confirms our predictions about which profile headings would go with which clusters. The teletype output is shown, with the 15 unique words underlined:

<u>Profile #1</u>	<u>Profile #2</u>	<u>Profile #3</u>	<u>Profile #4</u>
GOVERNMENT	MAGNETIC	TOPIC	<u>INDEXING</u>
<u>SERVICE</u>	<u>SIGNAL</u>	SEPARATION	<u>PROCESSING</u>
TECHNOLOGY	DIGITAL	COEFFICIENT	<u>DESCRIPTOR</u>
THINK	<u>MEMORY</u>	CORRESPONDENCE	ROBBERY
PROFESSION	ADDRESS	SYNONYM	<u>LANGUAGES</u>
SCIENTIFIC	WINDING	PROPERTY	<u>USED</u>
RESPONSIBILITY	<u>CIRCUIT</u>	CHARACTERISTIC	BASIS
COMMUNITY	IBM	REDUNDANCY	ACCORDING
<u>AGENCY</u>	SWITCH	<u>HOMOGRAPH</u>	GENERAL
LIBRARIAN	SEQUENCE	<u>TOKEN</u>	RESULTS
COUNTRY	<u>PERMANENT</u>	PHRASE	RULES
SOCIETY	WIRE	SUBSET	<u>PARTS</u>

You, the reader, comment that the documents whose lists cluster around profile #1 must have been written by habitual "think biggers," in contrast to those giving rise to the profile #2 words, that appear to talk about the nuts and bolts of computers. Your observation is approximately correct.

The rather strong indications of clustering tendencies encourage us to return to mode 1 of the matching function, assigning all of the 419 lists.

You inquire, "Why haven't we used mode 2?" The programmer explains that mode 2 is just an alternative way of screening out the ambivalent lists--the ones with low ratios of highest profile-score to next highest. With mode 2 the cutoff is in terms of the value of the ratio (i.e., use only lists having a ratio greater than 1.20), whereas mode 3 selects the n highest-ratio lists. Though mode 2 has its uses, it cannot be used to control profile scoring power the way we used mode 3.

It is advantageous to use mode 1 whenever the mode 3 control is not needed, because the full amount of the information content inherent in all the lists can be brought to bear on classification; in fact, we suspect that in the present run we need as much information as we can muster, if indeed the robbery report lists are being misclassified, as it looks like they are. You point out quite cogently that if a user of ALCAPP is working with new and unfamiliar information, "How is he supposed to know that the clustering is going badly?" The programmer quips, "Let him use word magic in selecting his indifferent starting distributions, and then he'll know definitely that it will go badly." Then he adds, "If you are using the program for business, you don't need to start with a random distribution. There are crude but effective single-pass sorting procedures that can give you a highly biased starting distribution--biased towards the clustering tendencies in the data. The iterative cycles just clean up the inevitable umpteen percent of misclassified items."

You encounter, "What if that umpteen percent includes all of the robbery report stuff?" The programmer says, "The way we started our run was just asking for trouble. Starting with a small subset of randomly, or indifferently, selected lists and setting up profiles from them is an open invitation for premature formation of some subcluster in the wrong place. All it takes is a couple of lists like 291 and 293 that just happen to be from the same subcluster. In a crude single-pass sort, all of the items will be equally involved, and no cluster or subcluster is given a preferential chance to get established, as in the case we're looking at."

We now present a table reflecting what you, the reader, see during the next several cycles. We use mode 1 and mode 3 more or less alternately, which has the effect of speeding up progress toward convergence.

<u>Cycle Number</u>	<u>Matching Mode</u>	<u>Number of Unique Words Duplicated</u>	<u>Correctly Assigned Starting Lists</u>	<u>Lists 287-295 with Profile #4</u>
4	1	15 out of 48	23 out of 40	All
5	3	18 " " "	23 " " "	5 out of 9
6	1	21 " " "	26 " " "	6 " " "
7	1	29 " " "	29 " " "	7 " " "
8	3	32 " " "	29 " " "	7 " " "
9	1	35 " " "	33 " " "	1 " " "
10	1	29 " " "	34 " " "	1 " " "
11	1	29 " " "	33 " " "	None

Several things are noticeable about these cycles. First, the strategy of occasional use of mode 3 is just one thing for dislodging the robbery report lists; by cycle 11 they are all assigned to their usual cluster, that of profile #2, which tends to be a catch-all for the miscellaneous material as well as a category for general topics in the computer field. Second, the other indicators of increasingly cohesive clusters appear to plateau after about cycle 7. The attainment of this improvement plateau ordinarily occurs on cycle 4 or 5, given more fortunate "indifferent" starting distributions. Its significance is that the majority, meaning 60 to 70 percent, of the lists have found their proper clusters and will remain assigned there in subsequent cycles. After this point the improvement gradient is relatively shallow for from 5 to 10 cycles, another 3 or 4 cycles after that involve only slight changes and lead up to convergence. The attainment of the plateau, however, is regarded as a more significant event than actual convergence because, as explained earlier, the half-dozen-or-so cycles leading up to convergence affects the fate of the ambivalent lists only. Needless to say, there are a lot of other things we might want to do with the ambivalent lists besides force them to "make up their minds" which category they want to be in.

Scrutinizing more closely the meanderings between clusters of our 40 starting lists, we find (by inspecting the raw distribution data that is too voluminous to be reproduced here) that 21 out of the 40 never change profiles after cycle 4. Between cycles 4 and 9, another 12 lists trickle into clusters to which they are destined to be firmly attached; the last 7 lists, if we let the program run beyond cycle 11 and converge, will "fast slosh" from profile to profile if only mode 1 is used (and this was the only mode available in ALCAPP's predecessor). If we now look at the profile-score ratios of these three groups, it will become clear why it was decided to program modes 2 and 3:

<u>Number of lists and classification behavior</u>	<u>Average score ratios, highest/second highest</u>	<u>Range of central two-thirds</u>
21 lists finding correct cluster by cycle 4	1.70	1.20 to 2.10
12 " " " " betw. cycles 4 & 9	1.37	1.20 to 1.60
7 " not " " " by cycle 9	1.10	All below 1.3

An ambivalent list usually has its ratio between 1.00 and 1.20 (though some in this range are not ambivalent). Of the 7 not properly classified lists (assuming the distribution from the original profiles as a standard of comparison), 5 had ratios below 1.20. Among the 33 well-classified lists occurred only two such ratios. Mode 3, then, serves two purposes: (1) to equalize profile scoring power in the early cycles, and (2) to prevent, in the approach to convergence, the ambivalent lists from contributing to profile generation and thus allowing the profiles to reflect, amplify, and perpetuate fast slosh.

We cannot take leave of this "live demonstration" of ALCAPP for the reader without showing how close "word magic" came to duplicating the lists of words used to derive the list numbers in the starting distribution. To save the need for page turning, we underline each word present in the selection we started with. The numbers after the words show original rank in uniqueness--even in this a healthy correlation shows. Here are the unique words that were printed out in the 9th cycle:

<u>Profile #1</u>		<u>Profile #2</u>		<u>Profile #3</u>		<u>Profile #4</u>	
<u>SCIENTIST</u>	2	<u>SIGNAL</u>	1	<u>MEANING</u>		<u>DOCUMENTS</u>	3
<u>SERVICE</u>	1	<u>MEMORY</u>	3	<u>COOCCURRENCE</u>	1	<u>PROBLEMS</u>	1
<u>SCIENCE</u>		<u>SIGN</u>	2	<u>COUNT</u>	3	<u>INDEXING</u>	2
<u>STUDENT</u>	3	<u>CIRCUIT</u>	6	<u>HOMOGRAPH</u>	8	<u>PROGRAMS</u>	5
<u>AGENCY</u>	6	<u>PULSE</u>	8	<u>COEFFICIENT</u>		<u>TECHNIQUES</u>	4
<u>PATENT</u>		<u>DIGITAL</u>		<u>EMPIRICAL</u>	5	<u>WORDS</u>	6
<u>PROFESSION</u>	9	<u>SEQUENCE</u>		<u>TOKEN</u>	11	<u>TERMS</u>	7
<u>EDITOR</u>	11	<u>CONDITION</u>		<u>CORPUS</u>		<u>SYSTEMS</u>	8
<u>CATALOG</u>	4	<u>SWITCH</u>	5	<u>MARON</u>	6	<u>PROCESSING</u>	9
<u>DISSEMINATION</u>	8	<u>SENSE</u>	11	<u>STRONG</u>	9	<u>LANGUAGES</u>	10
<u>ENGINEER</u>	5	<u>IMAGE</u>		<u>LABEL</u>		<u>USED</u>	
<u>EDUCATION</u>		<u>PARAMETER</u>		<u>MARKER</u>	12	<u>USING</u>	12

Part of the reason for the greater unique-word duplication in cycle 9 than in the neighboring cycles is that cycle 8 provides a mode-3 distribution--with the ambivalents excluded! In another run the day after this one ("this one" was of course a real run* made without you, the reader, present),

* We are saving the TTY output, dated June 29, 1966, for the benefit of the unduly skeptical.

the author succeeded in having the unique words for profile #4 emerge in identical rank order to the words for original profile #1, in spite of the fact that cluster membership was not exactly the same in both cases.

D. Prospects for Declining Costs and Continued Development

In being preoccupied with the long and arduous task of "bronc busting" the iterative process and mastering its idiosyncrasies, we on the automatic classification project almost failed to notice that the unit cost of our method had declined by a factor of ten, simply as a result of increased understanding of which operations are and are not essential in our procedure. No gains were made on our part as the result of changes to faster or better computers and peripheral equipment. It was on the basis of our more efficient program, ALCAPP, that I made the cost estimate of $1\frac{1}{2}$ cents per document at the very end of Section I.

But that charge rate for automatic classification, low as it is, is not one I would bother to defend or even compute more carefully, because it is almost certainly going to change--downward. Our factor-of-ten cost reduction was simply a by-product of our need as researchers for faster programs. What could we achieve in further reduction if we really worked at it? Of course, the biggest cost-reduction accomplishment of all was simply making the method work, with all its implications--analyzed in Section II-E--for the cheap processing of numbers of items in excess of 100,000. It would appear at this time that our most appropriate follow-up is to consolidate that gain.

Such activities as starting from random or indifferent starting distributions or reduplicating previous things identically are only for demonstration purposes. An in-use automatic classification system would have a no-nonsense sorting booster that would give a thoroughly biased (in the proper direction) starting distribution. This facility is in our plans. Ambivalent documents would not be allowed to tie up computer time in extra cycles needed for convergence; we hope to find ways of spotting them early and ways of dealing with them, once spotted.

Cost barriers, if any, will probably not be in the computer-time usage by the iterative process, but in the tasks that are required in getting material ready for storage and in putting the output to use. Fortunately, many agencies or individuals are currently planning to put large volumes of text in machine-readable form for purposes other than automatic classification. For these people, classification would be such a cheap by-product that if it were of any use at all to them, they would easily be able to afford it.

It may be redundant, but nevertheless surprising to many readers, to point out that individual pages in books contain more than enough information to permit their accurate classification by automatic means; no librarian

could quarrel with the appropriateness of this, since any kind of classification of page-sized stretches of text is clearly out of the question without computers. Robert Simmons, a fellow researcher, fed a number of segments from a psychology text to the obsolescent Ward grouping program (6), and was notably impressed with the results. The potential for access by professional persons to the books and journals in their own office is readily apparent. For some trained people with critical jobs, the value of the access could literally justify the cost of key-punching the contents of every article and notebook in the office. It is not an accident, by the way, that the author has derived many word lists from his own material.

No one who knows what we know--those of us who are closely familiar with the technical success of ALCAPP and with the possibilities of "direct proportion" automatic classification methods in general--can escape the conclusion that the "cost barrier" in automatic classification has been broken. It was because of this acute realization that I have emphasized so strongly in Section I the "intellectual resistance" factor. If I were a soothsayer, I would predict solemnly that automatic classification is not about to sweep the country this year or the next; but the reasons for this prediction being borne out will not be reasons of cost.

Mere cheapness or availability alone may not lead to use; for years in many oil fields, natural gas was burned off as unwanted waste. It may have been that many oil producers were aware of the waste, but also saw that, though the gas was quite cheap, the developmental investments required to benefit from this fact were too large: the cost of pipelines, of pressurized tanks, and of distribution facilities at the consumer end.

Automatic classification is almost in a parallel situation. User acceptance cannot be won through spreading the word that the world's literature may now be classified for $1\frac{1}{2}$ cents an item. Developmental investments both at the text-input end and the output end may turn out to be larger than anyone will think worthwhile; it is of course at the critical point of decision-making regarding such development that the "intellectual resistance" factor may continue to bottle up the genie.

We who work on automatic classification would probably be safe in assuming that beyond the cost barrier there is a user-acceptance barrier. Being beyond the cost barrier may not mean, in itself, that the user-acceptance barrier is more formidable; it may actually be less so. But it would be a mistake to expect the assistance of the middlemen (information specialists, librarians, etc.) who view the products and persuasions of the "computer colony" with suspicion, no doubt justifiably in many cases, and who are quite sure that however much computer people may understand computers, they do not understand users of documented information.

Convincing these middlemen is not the route to take. As Carter et al. (12) point out: "...Most librarians and the traditions of librarianship are grounded in the humanities rather than in technology. As a result, many policy makers in libraries tend to be very unsure of the potentials of modern technology..." To "unsure of the potentials" might have been added "and fearful of the blight": we have heard repeatedly--clothed in situation-specific terms--the sentiment that mechanization undermines human values, saps individuality, discourages craftsmanship, and dulls the mind. It is easy for some to forget that there would be no books and no libraries if it were not for that mindless 15th century technology known as "movable type."

We can be especially sure that the statistical approach in particular will be unwelcome, because of a conviction that reading and assimilating must be an intellectual matter, and, ergo, the ancillary functions of indexing, abstracting, classification, etc. There would be certainty that books and articles were not written for the purpose of having their words counted. It is difficult for me to resist translating the kind of thinking involved in these attitudes into a somewhat more familiar setting, like: "Human voices weren't meant to be changed into electric currents and bounced off the top of the sky. Besides, I don't hold with furniture that talks." We would not expect a person with such an opinion to have invested in RCA stock, since his conception of the "user" who supposedly would buy the talking contraption would be that of a solid citizen like himself.

Developers of such things as automatic classification must assiduously cultivate their own model of the user, and eventually hope to bypass the middleman in pursuit of user acceptance. Such a user-oriented way of thinking is not difficult today, since many user studies and summaries of user studies are available. In addition, we are users, you and I and the fellow across the hall; since we know ourselves and our interactions with information, we cannot be said to be totally ignorant of the problems of users. Furthermore, we cannot afford to be biased in our notions about the user; the user-acceptance barrier will not yield unless it is studied with an open mind.

The investment required to surmount the user-acceptance barrier will be on the output side of the automatic classification process. As it stands, a great part of the investment has been made. Many new types of output hardware are now in use and becoming cheaper: cathode-ray display scopes with light pens, plotters, remote teletypes, and so on, all having as-yet unrealized potentials in bringing the user closer to information.

This author has done more than his share of thinking about how to make computer output palatable to users, thinking--for example--in terms of displays from the very outset (13). Many modes of involving intervention

of human intelligence in the form of editing have been described (14). Assumptions made about the user, particularly about the importance of physical proximity to information access (15), have been subsequently verified by user studies (see Section 8, Ref. 12).

Finally, of course, as indicated by Figure I in Section II, our present automatic classification project has steadily been concerned with formats that the output might assume, and has realistically faced up to the fact that after all the millions of documents are pigeonholed with accuracy and economy, the whole business still has to be revealed to and comprehended by the user himself. Our slogan for now is: No Classification Without Representation.

REFERENCES

1. Bruner, J. S., Goodnow, J. A. and Austin, G. A. A Study of Thinking. New York: John Wiley and Sons, Inc., 1956.
2. Doyle, L. B. Is automatic classification a reasonable application of statistical analysis of text? Journal of the Association for Computing Machinery, 1965, 4, pp. 473-489.
3. Borko, H. Measuring the reliability of subject classification by men and machines. American Documentation, 1964, 4, pp. 268-273.
4. Doyle, L. B. The microstatistics of text. Information Storage and Retrieval, 1963, 4, pp. 189-214.
5. Doyle, L. B. Some compromises between word grouping and document grouping. Statistical Association Methods for Mechanized Documentation (Symposium Proceedings, Washington 1964), National Bureau of Standards Miscellaneous Publication 269, pp. 15-24.
6. Ward, J. H., Jr. and Hook, M. E. Application of a hierarchical grouping procedure to a problem of grouping profiles. Educational and Psychological Measurement, 1963, 23, pp. 69-92.
7. Zipf, G. K. Human Behavior and the Principle of Least Effort. Cambridge, Mass.: Addison-Wesley Press, Inc., 1949.
8. Williams, J. H. Results of classifying documents with multiple discriminant functions. Statistical Association Methods for Mechanized Documentation (see Ref. 5, above).
9. Forgy, E. W. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. Presented June 22, 1965, at the AAAS-Biometric Society Meeting, Riverside, Calif.
10. Ball, G. H. Data analysis in the social sciences: what about the details? Proceedings of the Fall Joint Computer Conference. Washington, D.C.: Spartan Books, 1965, pp. 533-59.
11. Doyle, L. B. Re-expression in standardized code to improve the automatic classifiability of text items. SDC document TM-2213, February 1965.
12. Carter, L. F., et al. Recommendations for National Document Handling Systems in Science and Technology. SDC document TM-WD-213/001/00, September 1965. Available from Clearinghouse for Federal Scientific and Technical Information, No. PB 163 267.
13. Doyle, L. B. Semantic road maps for literature searchers. Journal of the Association for Computing Machinery, 1961, 4, pp. 553-578.
14. Doyle, L. B. Expanding the editing function in 1 data processing. Communications of the Association for Computing Machinery, April 1965, pp. 238-243.
15. Doyle, L. B. How to plot a breakthrough. SDC document SP-1492, December 1963.

DOCUMENT CONTROL DATA		
1. ORIGINATING ACTIVITY (Corporate author)		2. REPORT SECURITY CLASSIFICATION
System Development Corporation Santa Monica, California		Unclassified
3. REPORT TITLE		
Breaking The cost Barrier in Automatic Classification		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)		
5. AUTHOR(S) (Last name, first name, initial)		
Doyle, L. B.		
6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
1 July 1966	64	15
8a. CONTRACT OR GRANT NO	8b. ORIGINATOR'S REPORT NUMBER(S)	
AF 19(628)-5166, Prototype Library Project, for Rome d. PROJECT NO Air Development Center	SP-2516	
c.	9a. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d.		
10. AVAILABILITY/LIMITATION NOTICES		
Distribution of this document is unlimited		
11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY	
13. ABSTRACT		
<p>A low-cost automatic classification method is reported that uses computer time in proportion to $N \log N$, where N is the number of information items and the base is a parameter. Some barriers besides cost are treated briefly in the opening section, including types of intellectual resistance to the idea of doing classification by content-word similarity. The second section explains the basic processes of document grouping by similarity, and discusses the advantages of the reported method over methods commonly experimented with. The operation of an iterative procedure using word profiles to progressively improve the grouping of content-word lists is described. Then some possible applications aside from document classification are enumerated. The final section begins by presenting theoretical underpinnings that explain the form taken by the components of the method. An account of the struggle to make the method work is sketched, followed by a cycle-by-cycle description of a feasibility demonstration. The conclusion states that mere cheapness is not enough and analyzes what researchers and developers might have to do before user acceptance of automatic classification can be assured.</p>		

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Automatic Classification Document Grouping Similarity Content-Word Lists Costs						

INSTRUCTIONS

1. **ORIGINATING ACTIVITY:** Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization (corporate author) issuing the report.

2a. **REPORT SECURITY CLASSIFICATION:** Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. **GROUP:** Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. **REPORT TITLE:** Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. **DESCRIPTIVE NOTES:** If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. **AUTHOR(S):** Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. **REPORT DATE:** Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. **TOTAL NUMBER OF PAGES:** The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. **NUMBER OF REFERENCES:** Enter the total number of references cited in the report.

8a. **CONTRACT OR GRANT NUMBER:** If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. **PROJECT NUMBER:** Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. **ORIGINATOR'S REPORT NUMBER(S):** Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. **OTHER REPORT NUMBER(S):** If the report has been assigned any other report numbers (either by the originator or by the sponsor), also enter this number(s).

10. **AVAILABILITY/LIMITATION NOTICES:** Enter any limitations on further dissemination of the report, other than those

imposed by security classification, using standard statements such as:

- (1) "Qualified requesters may obtain copies of this report from DDC."
- (2) "Foreign announcement and dissemination of this report by DDC is not authorized."
- (3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."
- (4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."
- (5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. **SUPPLEMENTARY NOTES:** Use for additional explanatory notes.

12. **SPONSORING MILITARY ACTIVITY:** Enter the name of the departmental project office or laboratory sponsoring (paying for) the research and development. Include address.

13. **ABSTRACT:** Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as (TS), (S), (C), or (U).

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. **KEY WORDS:** Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.